

せかちゃん

世界一わかりやすい  
情報科チャンネル

授業の復習や共通テスト対策に  
高校情報Ⅰ授業ノート

ExcelVBA 編



## 授業ノートの利用にあたって



はい、こんにちは！ せかチャンの菅原です。  
授業ノートを手にしてくれてありがとうございます！  
今、たくさんの高校生や大学生、社会人のみなさんが  
せかチャンの動画で情報技術やプログラミングを学んでいます。  
「分かるから楽しい！」そんな声も多いです。  
1人より誰かと一緒のほうが不思議とがんばれる。  
あなたも、僕たちと一緒に楽しく勉強しませんか？

### 1. 【せかチャンの動画さえ見ておけば大丈夫！】な理由

情報Ⅰ科目は13種類（2024年4月現在）の教科書が刊行されています。実は、教科書によって目次の構成やどこまで深く解説されているかが全然違うんですよね。なので、あなたも

「せっかく参考書を買ったのに、学校の授業で習った順番と全然違う…」

「模試で教科書に載ってない問題が出たんだけど。こんなの解けるわけないじゃん！」

このような経験があるかもしれません。

せかチャンでは、主要4社の教科書と情報科教員向けの研修資料をもとに授業動画を作ってきました。すべての教科書をカバーできるように作ったため、あなたの教科書では解説されていないキーワードが出てくるかもしれません。ただ、動画内で解説している内容はすべて重要なものばかりです。基本から解説しているので、ぜひ理解しておいてくださいね。

この授業ノートを手元に置いて、動画を見ながらメモをしたり確認クイズを解いたりして理解を深めましょう。せかチャンの動画とこの授業ノートで、ぜひ情報Ⅰを得意科目にしてください！

### 2. 授業ノートの使い方

授業の予習・復習や定期テスト対策、大学入学共通テスト対策の第一歩にオススメ！最初からひと通り勉強しても良いですし、各Lessonの情報を参考に優先度をつけて勉強するのも効率的ですね。

学ぶテーマ

重要度

難易度

Lesson 1

情報とは

0:00:33

動画の開始時間と QR コード

※重要度と難易度は3段階

### 3. 利用にあたって

せかチャンの動画や教材は、情報技術を学ぶすべての人たちに役立ててほしいと願って制作しています。高校の先生方から「授業で使っても大丈夫？」「板書やプログラムを印刷配布してよい？」のような質問を頂きますが、ぜひ生徒さんのために自由に使ってください！

営利目的の場合にのみ、ご連絡を頂ければ大丈夫です。



## 目次

授業ノートの利用にあたって .....	2
目次 .....	3
1章 情報社会の問題解決 .....	4
2章 コミュニケーションと情報デザイン .....	15
3章 コンピュータとプログラミング .....	34
4章 情報通信ネットワークとデータの活用 .....	65
付録 スクショタイムの画像一覧 .....	87



【1章】共通テスト対策はここから始めよう！

 <https://youtu.be/IDqy3b1ICiQ>

動画時間：1:33:09

学習項目：10



### 1章のゴール

- 情報やメディアの特性を理解しよう
- 問題を発見して解決する流れを理解して、発想法や計画管理のツールを活用できるようになる
- 情報に関する法律や制度、セキュリティ技術を理解しよう
- 情報技術が人や社会に果たす役割や影響、情報モラルについて理解して、情報や情報技術を適切に活用できるようになる

### ▶ キーワード

- |                                     |                                       |                                      |
|-------------------------------------|---------------------------------------|--------------------------------------|
| <input type="checkbox"/> 情報         | <input type="checkbox"/> 特許権          | <input type="checkbox"/> 不正アクセス      |
| <input type="checkbox"/> データ        | <input type="checkbox"/> 実用新案権        | <input type="checkbox"/> ソーシャル       |
| <input type="checkbox"/> メディア       | <input type="checkbox"/> 意匠権          | <input type="checkbox"/> エンジニアリング    |
| <input type="checkbox"/> 表現メディア     | <input type="checkbox"/> 商標権          | <input type="checkbox"/> マルウェア       |
| <input type="checkbox"/> 伝達メディア     | <input type="checkbox"/> 著作権          | <input type="checkbox"/> コンピュータウィルス  |
| <input type="checkbox"/> 記録メディア     | <input type="checkbox"/> 著作者人格権       | <input type="checkbox"/> Society 5.0 |
| <input type="checkbox"/> ブレーンストーミング | <input type="checkbox"/> 著作隣接権        | <input type="checkbox"/> IoT         |
| <input type="checkbox"/> PERT 図     | <input type="checkbox"/> 個人情報         | <input type="checkbox"/> ビッグデータ      |
| <input type="checkbox"/> ガントチャート    | <input type="checkbox"/> 個人情報保護法      | <input type="checkbox"/> AI（人工知能）    |
| <input type="checkbox"/> 知的財産権      | <input type="checkbox"/> 情報セキュリティポリシー |                                      |
| <input type="checkbox"/> 産業財産権      | <input type="checkbox"/> サイバー犯罪       |                                      |



## ▶ ホワイトボード

▶ 情報とは

① なぜ学ぶの？

便利だけど 凶器にもなる! → 情報技術も同じ → 便利 & 楽しい! 周りを助けよう!

① 情報って何？

ある物事や事情の報告で意思決定の判断基準になる

↑ 意味・解釈 ↓ 分析・体系化

観測や調査などで得られる客観的事実

情報が役立つ形で蓄積されたもの

② 情報の特性

モノとしての がない

だから

① \_\_\_\_\_ : 伝えてもなくなる  
② \_\_\_\_\_ : 簡単にコピーできる  
③ \_\_\_\_\_ : 瞬時に伝わる

## ▶ 確認テスト

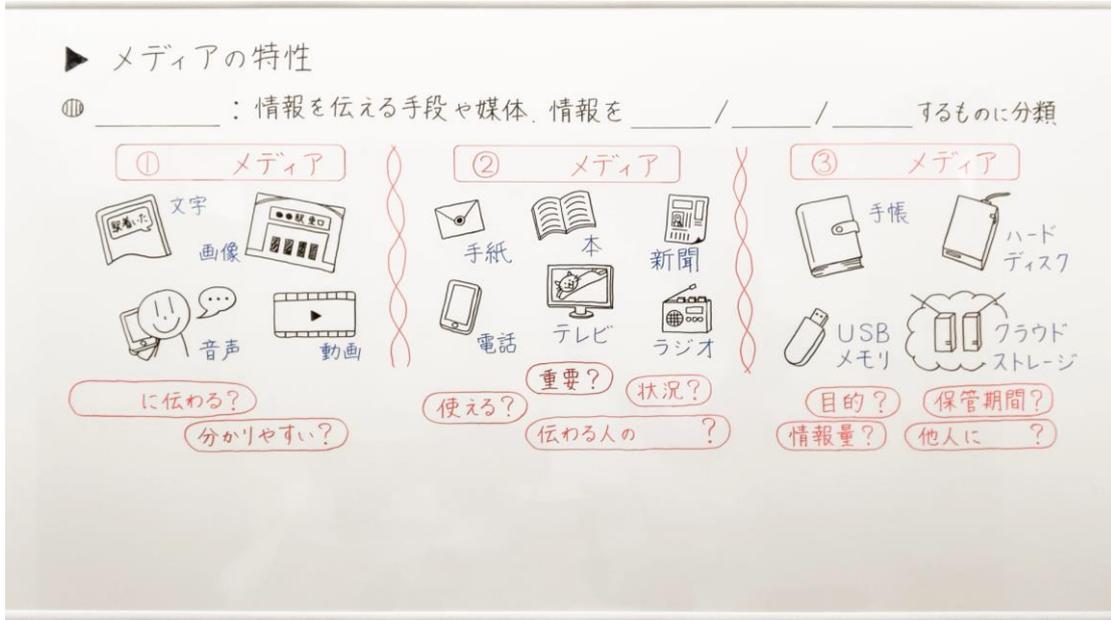
画面を止めて考えてみよう!

- ① ある物事の内容や事情の報告で、意思決定の判断材料になるものを(① \_\_\_\_\_)とよぶ。(①)は観測や調査などで得られる(② \_\_\_\_\_)に意味や解釈が加えられてつくられる。(①)が役立つ形で蓄積されたものを(③ \_\_\_\_\_)という。
- ② (①)にはモノとしての(④ \_\_\_\_\_)がない。そのため、伝えてもなくなる、簡単にコピーできる、瞬時に伝わるといった特性がある。

(確認テストの回答は次ページに掲載)



## ▶ ホワイトボード



## ▶ 確認テスト

画面を止めて考えてみよう!

情報を伝えるためのメディアは、その特性によって次の3つに分類される。

- ・(① \_\_\_\_\_)メディア:情報を表現するためのメディア
- ・(② \_\_\_\_\_)メディア:情報を物理的に伝えるためのメディア  
特にテレビや新聞のように、不特定多数に発信する(②)メディアを(③ \_\_\_\_\_)という
- ・(④ \_\_\_\_\_)メディア:情報を長く保管するためのメディア

①情報 ②データ ③知識 ④形



## ▶ ホワイトボード

▶ 問題解決 - 発想法のツール -

① 問題の  
現実と理想のギャップに気付く

② 問題の  
問題を具体化してゴールを設定

③ 解決案の検討  
情報を収集・分析して  
解決案を考える

④ と  
計画を立てて実行する

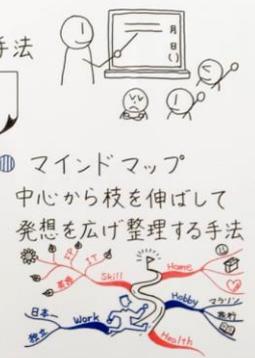
⑤ 振り返り  
結果を評価して次に活かす

④ ブレインストーミング  
複数の人で多くのアイデアを出す手法

◎ 質より	◎ でユニーク
◎ 他人に	X &

④ KJ法  
カードでアイデアを整理する手法

④ マインドマップ  
中心から枝を伸ばして  
発想を広げ整理する手法



## ▶ 確認テスト

画面を止めて考えてみよう!

問題を明確化したり解決案を検討したりする手法として、複数  
 の人で多くのアイデアを出す(① \_\_\_\_\_)  
 がある。(①)ではアイデアの質より(② \_\_\_\_\_)を優先して、自由  
 でユニークな意見を歓迎して話し合いを進める。

カードを使ってアイデアを整理する手法として(③ \_\_\_\_\_)  
 )がある。また、中心から枝を伸ばすように発想を広げながら整  
 理する手法である(④ \_\_\_\_\_)も便利である。

①表現 ②伝達 ③マスメディア ④記録



## ▶ ホワイトボード

▶ 問題解決 - 計画管理のツール -

① 問題の発見  
現実と理想のギャップに気付く

② 問題の明確化  
問題を具体化してゴールを設定

③ 解決案の検討  
情報を収集・分析して  
解決案を考える

④ 計画と実行  
計画を立てて実行する

⑤ 振り返り  
結果を評価して次に活かす

```

graph TD
    A["A 1日  
内容決定"] --> D["D 3日  
プレ検討"]
    A --> B["B 1日  
お店検討"]
    B --> C["C 1日  
お店予約"]
    C --> F["F 5日  
プレ作成"]
    D --> E["E 1日  
プレ買物"]
    E --> F
    F --> G["G 1日  
当日"]
  
```

● PERT図  
作業間の \_\_\_\_\_ を表現

● ガントチャート  
作業内容と \_\_\_\_\_ の関係を表現

A 内容決定	全員	5	1	2	3	4	5	6	7
B お店検討	菅原・梅井								
C お店予約	:								
D プレ検討	田原・中川								
E プレ買物	:								
F プレ作成	全員								

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 問題の解決案を行動を起こすときには、計画を立てて可視化することが大切である。作業間の順序関係を分かりやすく矢印で表現した図を(① \_\_\_\_\_)といい、重要な作業を判断するのに便利である。作業内容と期間の関係を表す(② \_\_\_\_\_)は、計画を可視化するのに便利である。
- ② Plan(計画)、Do(実行)、Check(評価)、Action(改善)を繰り返し改善を続ける手法を(③ \_\_\_\_\_)とよぶ。

①ブレインストーミング ②量 ③KJ法 ④マインドマップ



## ▶ ホワイトボード

▶ 知的財産権

① \_\_\_\_\_ : 知的な創作活動から生まれた価値を保護  
 ② \_\_\_\_\_ : ビジネス → 特許庁への出願で発生

権 [発明・20年]  
 権 [アイデア・10年]

権 [デザイン・25年]  
 権 [ブランド・10年]

② \_\_\_\_\_ : 文化・芸術 → 創作時に発生

作者の権利

作者人格権 著作権(財産権)

公表権 複製権  
 氏名表示権 上映権  
 同一性保持権 公衆送信権

実演家 放送事業者

録画権など 複製権など

他人の著作物を利用するとき

原則 著作権の \_\_\_\_\_ を得る  
 例外 保護期間(70年)を経過  
 私的利用のコピー  
 教育機関でのコピー  
 非営利目的の上演 など

引用のルール

① \_\_\_\_\_ がある ② 全体の \_\_\_\_\_  
 ③ 他に \_\_\_\_\_ 可 ④ \_\_\_\_\_ の明記

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 知的な創作活動から生まれた価値を保護する権利を(① \_\_\_\_\_ )という。(①)はビジネスに関する(② \_\_\_\_\_ )と、文化に関する(③ \_\_\_\_\_ )に分けられる。
- ② (②)には高度な発明を扱う(④ \_\_\_\_\_ )や、早期に実施できるアイデアを扱う(⑤ \_\_\_\_\_ )などがある。(③)は小説や音楽などの創作者に与えられる権利で、著作物を利用する場合は原則として(⑥ \_\_\_\_\_ )を得る必要がある。



## ▶ ホワイトボード

▶ 個人情報

個人を識別できる情報

取り扱い方法を規定

個人情報取扱事業者の義務

個人を識別できる情報

氏名 ○○○○ 性別 ○

住所 ○○○○

生年月日 ○-○-○

年齢、TEL、メールアドレス、職業 など

他人に知られたくない情報

好み、購入・閲覧履歴 など

基本方針：価値観

対策基準：ルール

実施手順：具体的な方法

顔や姿の撮影を拒む権利

情報漏洩（攻撃、紛失）

流出

自分のミス、記入・入力

→ させた人：懲戒解雇、損害賠償、窃盗罪 など

→ された人：不正、迷惑 など

GOOGLE

プライバシーポリシー

## ▶ 確認テスト

画面を止めて考えてみよう！

- ① 生存する個人を識別できる情報を(① \_\_\_\_\_)とい  
い、氏名や住所、性別、生年月日などがある。好みや購入履歴  
のように他人に知られたくない情報は(② \_\_\_\_\_)  
といい、取り扱いには配慮が必要である。
- ② (③ \_\_\_\_\_)には(①)取扱事業者の義務など  
が規定されている。情報セキュリティに関する方針や行動指針  
を(④ \_\_\_\_\_)で定める事業者も多い。

①知的財産権 ②産業財産権 ③著作権 ④特許権 ⑤実用新案権 ⑥許諾



## ▶ ホワイトボード

▶ サイバー犯罪

● \_\_\_\_\_ : コンピュータやネットワークを悪用した犯罪

① 不正アクセス禁止法違反  
ID・パスワードを不正入手・使用

② コンピュータ・電磁的記録対象犯罪  
コンピュータの不正利用やデータの改ざん・破壊

③ ネットワーク利用犯罪  
詐欺・誹謗中傷 など

	詐欺	詐欺	詐欺
<p>① メール受信/広告表示</p> <p>② 偽リンクをクリック</p> <p>③ ID・パスワード等入力</p> <p>↓</p> <p>個人情報の売買、不正送金、物品購入、不正ログイン、情報詐取</p> <p>★ 迷惑メールフィルター形式で受信通知</p>	<p>① web閲覧/メール受信</p> <p>② リンクをクリック</p> <p>③ 契約成立の画面表示</p> <p>↓</p> <p>あわてて入金 → 再要求 TEL → 個人情報詐取 救済業者 → 2次被害</p> <p>★ 安易にクリックしない、不当な請求を信じない、応じない、連絡しない</p>	<p>① ショッピングサイト閲覧</p> <p>② 注文・送金</p> <p>↓</p> <p>模倣品が届く 何も届かない → 返金不可</p> <p>★ 注文前に確認!</p>	<p>① 情報</p> <p>② 方法</p> <p>③</p> <p>④</p>

## ▶ 確認テスト

画面を止めて考えてみよう!

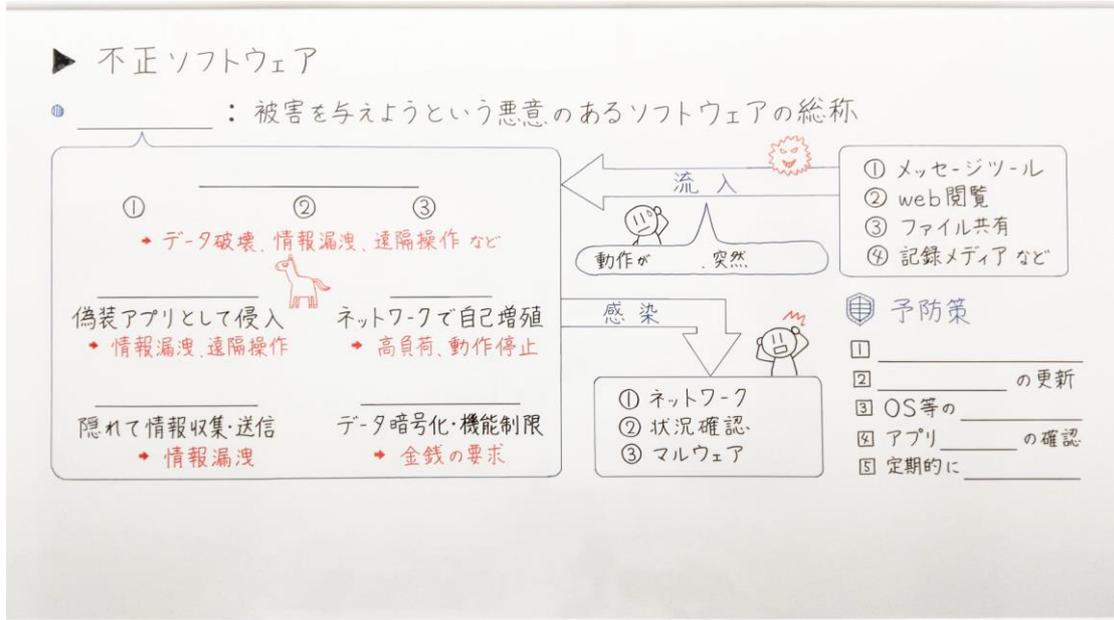
コンピュータやネットワークを悪用した犯罪のことを( ① \_\_\_\_\_ )という。

特にネットワークを介した犯罪には、メールや広告から偽サイトに誘導して個人情報を盗む( ② \_\_\_\_\_ )詐欺やリンクのクリックを契約成立と見せかけて架空請求をする( ③ \_\_\_\_\_ )詐欺、webサイトでの購入者から不正に送金を受けたり模倣品を送ったりする( ④ \_\_\_\_\_ )詐欺などがある。

- ①個人情報 ②プライバシー ③個人情報保護法 ④情報セキュリティポリシー



## ▶ ホワイトボード



## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 悪意のあるソフトウェアの総称を(① \_\_\_\_\_)という。(①)には自己伝染・潜伏・発病のうち1つ以上の機能をもつ(② \_\_\_\_\_)や偽装アプリとしてインストールされる(③ \_\_\_\_\_)などがある。
- ② (①)の感染を予防するには、(④ \_\_\_\_\_)の導入や、OSやソフトウェアを定期的に(⑤ \_\_\_\_\_)することが有効である。

①サイバー犯罪 ②フィッシング ③ワンクリック ④ショッピング



## ▶ ホワイトボード

▶ 情報モラル

① 適切な \_\_\_\_\_ (情報社会に対する行動や考え方) をもって楽しく便利に!

① すべての通信は \_\_\_\_\_

② \_\_\_\_\_ の情報に囲まれる

③ 正義感の \_\_\_\_\_ や \_\_\_\_\_

① \_\_\_\_\_ や \_\_\_\_\_ は大丈夫?

② \_\_\_\_\_ にいてもその言葉で話せる?

③ 発信内容への \_\_\_\_\_

情報機器の使いすぎによる \_\_\_\_\_ やスマホ/ネット \_\_\_\_\_ に注意!

## ▶ 確認テスト

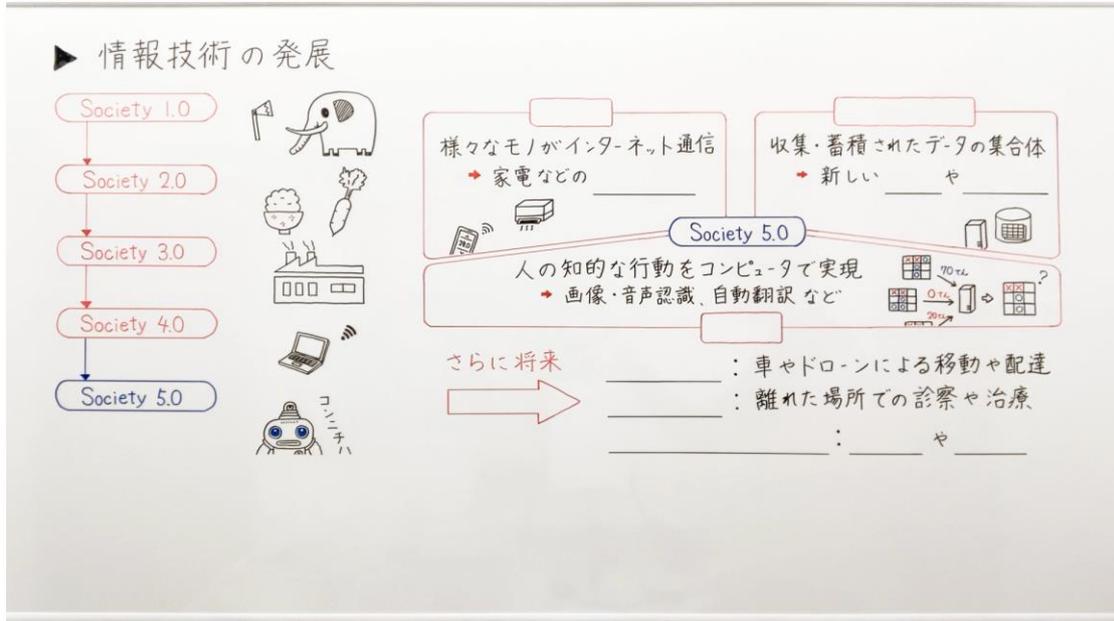
画面を止めて考えてみよう!

- ① 情報社会に対する適切な考え方や行動を情報(① \_\_\_\_\_)とよぶ。インターネットの通信では発信元を特定することが可能であり、インターネットでのコミュニケーションも日常における一般的な(①)と同じと考えるべきである。
- ② 情報機器やインターネットを長時間使用することで、目の疲れなどの(② \_\_\_\_\_)や、スマホ・ネット・ゲームなどの(③ \_\_\_\_\_)になることもあるので気をつけたい。

- ①マルウェア ②コンピュータウイルス ③トロイの木馬  
④ウイルス対策ソフトウェア ⑤アップデート



## ▶ ホワイトボード



## ▶ 確認テスト

画面を止めて考えてみよう!

工業社会や情報社会を経て、これから新しい社会である(① \_\_\_\_\_)の時代へ転換するといわれている。

(①)では様々なモノがインターネットでつながる(② \_\_\_\_\_)の環境が広がり、(②)などから多種多様な(③ \_\_\_\_\_)が収集・蓄積されていく。さらに、(③)を(④ \_\_\_\_\_)を通して分析・学習することによって、将来人間の仕事の多くを代替できるようになるといわれている。

- ①モラル
- ②VDT 障害
- ③依存症

# コミュニケーションと情報デザイン



【2章】情報技術の原理原則や仕組みを理解しよう

 [https://youtu.be/D-hrd\\_ugBos](https://youtu.be/D-hrd_ugBos)

動画時間 : 3:07:16

学習項目 : 18



## 2章のゴール

- 情報がどのようなデジタル技術で表現されているのかを知り、情報技術の原理原則を理解しよう
- メディアの発達によりコミュニケーションがどのように変化してきたのかを知ろう
- 情報デザインの考え方や技法を身につけて、効果的なコミュニケーションができるようになるろう

## ▶ キーワード

- |  |                                   |                                     |
|--|-----------------------------------|-------------------------------------|
| <input type="checkbox"/> アナログ          | <input type="checkbox"/> 光の三原色    | <input type="checkbox"/> メディア       |
| <input type="checkbox"/> デジタル          | <input type="checkbox"/> 色の三原色    | <input type="checkbox"/> ソーシャルメディア  |
| <input type="checkbox"/> 2進数・10進数・16進数 | <input type="checkbox"/> 解像度      | <input type="checkbox"/> 情報デザイン     |
| <input type="checkbox"/> 補数            | <input type="checkbox"/> 画素       | <input type="checkbox"/> 抽象化        |
| <input type="checkbox"/> 固定小数点数        | <input type="checkbox"/> 階調       | <input type="checkbox"/> 可視化        |
| <input type="checkbox"/> 浮動小数点数        | <input type="checkbox"/> フレーム     | <input type="checkbox"/> 構造化        |
| <input type="checkbox"/> 誤差            | <input type="checkbox"/> フレームレート  | <input type="checkbox"/> ユニバーサルデザイン |
| <input type="checkbox"/> 文字コード         | <input type="checkbox"/> 圧縮       | <input type="checkbox"/> アクセシビリティ   |
| <input type="checkbox"/> 標本化           | <input type="checkbox"/> 展開 (解凍)  | <input type="checkbox"/> ユーザビリティ    |
| <input type="checkbox"/> 量子化           | <input type="checkbox"/> ランレングス圧縮 |                                     |
| <input type="checkbox"/> 符号化           | <input type="checkbox"/> 通信       |                                     |



## ▶ ホワイトボード

▶ アナログとデジタル

① [ \_\_\_\_\_ : 連続した量( \_\_\_\_\_ )で表現すること. analogy (類似)が語源  
 \_\_\_\_\_ : とびとびの値( \_\_\_\_\_ )で表現すること. digit (指)が語源

② コンピュータは情報を \_\_\_\_\_ と \_\_\_\_\_ で表す

 電圧 高/低	 電流 ○/×
 磁気 N/S	 反射光 向き

③ デジタルデータの特徴

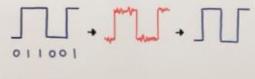
○ 利点	△ 欠点
(1) 様々な情報を _____ に扱える!	(1) 細かい情報が _____
(2) _____ や _____ がカンタン!	(2) _____ な理解が難しい
(3) 情報を _____ できる!	

Diagram showing a sine wave (analog) and a square wave (digital) connected by a double-headed arrow with blank spaces for labels.

## ▶ 確認テスト

### 画面を止めて考えてみよう!

1. 時間のように連続する情報を、時計の針の位置のような連続量で表現することを(① \_\_\_\_\_)という。一方、離散量で表現することを(② \_\_\_\_\_)という。
2. コンピュータは情報を(③ \_\_\_\_\_)と(④ \_\_\_\_\_)で表す
3. デジタル化によって、文字や静止画、動画などの情報を(⑤ \_\_\_\_\_)的に扱えるようになる。デジタルでは多少のノイズが混じっても情報を元通りに(⑥ \_\_\_\_\_)できる。

①Society 5.0 ②IoT ③ビッグデータ ④AI (人工知能)



## ▶ ホワイトボード

▶ 情報量

① デジタルでは数値・文字・音声・静止画・動画などを0と1の組合せ( \_\_\_\_\_ )で表現できる

<pre>       0      / \     0   1    / \ / \   0  1 0  1  / \ / \ / \ 0  1 0  1 0  1 </pre>	000 ↔ A	} 3桁は ( ) 通り
	001 ↔ B	
	010 ↔ C	
	011 ↔ D	
	100 ↔ E	
	101 ↔ F	
	110 ↔ G	
	111 ↔ H	

② 1桁の情報量 = | \_\_\_\_\_ [bit] → 2通り  
8ビット = | \_\_\_\_\_ [B] → 2<sup>8</sup>(256)通り

③ 大きい情報量を表現するのに \_\_\_\_\_ を使う

キロ	KB = 1024B	④ 情報量はいくつ? (1) 原稿テキスト(2800字) → 7.34 _____ (2) サムネイル画像 → 2.20 _____ (3) 動画(528) → 1.06 _____
メガ	MB = 1024KB	
ギガ	GB = 1024MB	
テラ	TB = 1024GB	

## ▶ 確認テスト

画面を止めて考えてみよう!

1. 情報量の最小単位で、0と1の1桁分の情報量のことを (① \_\_\_\_\_) という。また、8ビットをまとめて1 (② \_\_\_\_\_) といい、(③ \_\_\_\_\_) 通りの情報を表現することができる。

2. 次の情報量の単位を、小さいものから順に並べよう  
KB TB bit MB B GB

- ①アナログ ②デジタル ③0 ④1 (③④は順不同) ⑤統合 ⑥再現



## ▶ ホワイトボード

▶ 10進数と2進数

10進数	2進数
0	0
1	1
2	
3	
4	
5	
6	
7	
8	
9	
10	

① \_\_\_\_\_ : 10になると桁が進む数

$$365 = 3 \times \underline{\quad} + 6 \times \underline{\quad} + 5 \times \underline{\quad}$$

$$= 3 \times \underline{\quad} + 6 \times \underline{\quad} + 5 \times \underline{\quad}$$

▶ この表記法を \_\_\_\_\_、10を \_\_\_\_\_

② \_\_\_\_\_ : 2になると桁が進む数

$$9 = \underline{\quad} \times 2^3 + \underline{\quad} \times 2^2 + \underline{\quad} \times 2^1 + \underline{\quad} \times 2^0$$

$$\quad \quad \quad (8) \quad \quad (4) \quad \quad (2) \quad \quad (1)$$

▶ \_\_\_\_\_ は \_\_\_\_\_

10進数の9      2進数の1001

【 \_\_\_\_\_ は覚えると便利! 】

[Q1] 110010<sub>(2)</sub>を10進数に変換すると?

[Q2] 150<sub>(10)</sub>を2進数に変換すると?

## ▶ 確認テスト

画面を止めて考えてみよう!

1. 普段使っている数値は0~9を用いた(① \_\_\_\_\_)進数で表現されているが、コンピュータでは0と1のみを用いた(② \_\_\_\_\_)進数が使われている。
2. 2進数では、1桁上がるごとに各桁の(③ \_\_\_\_\_)が(④ \_\_\_\_\_)倍になる。
3. 2進数の10010110を10進数に変換しよう
4. 10進数の50を2進数に変換しよう

①ビット ②バイト ③256 2. bit, B, KB, MB, GB, TB



## ▶ ホワイトボード

▶ 16進数

- 2進数は \_\_\_\_\_ が多くなる
- 見やすくするために \_\_\_\_\_ を使う

Q1 365<sub>(10)</sub> を16進数に変換すると?  
 $365 = \_ \times 16^3 + \_ \times 16^2 + \_ \times 16^1 + \_ \times 16^0$   
 365<sub>(10)</sub> → \_\_\_\_\_

10進	0	...	9	10	11	12	13	14	15	16	17	18
16進	0	...	9	A	B	C	D	E	F	10	11	12
2進	0	...							1111	10000	10001	10010

Q2 110010<sub>(2)</sub> を16進数に変換すると?  
 Q3 4D<sub>(16)</sub> を2進数・10進数に変換すると?

- 2・16進数は \_\_\_\_\_ のタイミングが一緒!

## ▶ 確認テスト

画面を止めて考えてみよう!

- 2進数は桁数が多くなるので、下位から4桁ずつに区切って  
 (① \_\_\_\_\_)進数で表現されることがある。(①)進数では0  
 ~9の数字と(② \_\_\_\_\_)~(③ \_\_\_\_\_)の英字を使う。
- 16進数では、1桁上がるごとに各桁の(④ \_\_\_\_\_)が  
 (⑤ \_\_\_\_\_)倍になる。
- 16進数の16Dを10進数に変換しよう
- 2進数の1001101を16進数に変換しよう

①10 ②2 ③重み ④2 3. 150 4. 110010



## ▶ ホワイトボード

▶ 2進数の加減算と補数

① 2進数の加減算  
 \_\_\_ で繰り上がり(下がり)する

$$\begin{array}{r} 5+3 \\ 0101 \\ +)0011 \\ \hline \end{array}$$

$$\begin{array}{r} 5-3 \\ 0101 \\ -)0011 \\ \hline \end{array}$$

② \_\_\_\_\_ : その数を基準となる数から引いた数  
 \_\_\_\_\_ を表現できる

0011(3)の補数は?

基準の値(最大値+1)

$$\begin{array}{r} 10000 \\ -)0011 \\ \hline 1101 \end{array}$$

補数の求め方

① 0と1を  
(← の補数)

② 1を足す  
(← の補数)

$$\begin{array}{r} 5-3 \\ 0101 \\ +)1101 \\ \hline \end{array}$$

① 2の補数を足す  
(+10000 - 0011)

② 繰り上がりを切捨て  
(-10000)

→ 減算を \_\_\_\_\_ と \_\_\_\_\_ だけで実現!

## ▶ 確認テスト

画面を止めて考えてみよう!

1. コンピュータでは負の数を表現するときに(① \_\_\_\_\_ )  
 を使う。(①)を求めるには、各桁の0と1を(② \_\_\_\_\_ )  
 して(③ \_\_\_\_\_ )を足せばよい。
2. (①)を使えば減算が(④ \_\_\_\_\_ )と(⑤ \_\_\_\_\_ )  
 ができるため、演算用の回路をよりシンプルにできる。
3. 0101<sub>(2)</sub> の補数を求めよう
4. 1101<sub>(2)</sub> - 0101<sub>(2)</sub> を計算しよう

①16 ②A ③F ④重み ⑤16 3. 365 4. 4D



## ▶ ホワイトボード

▶ 2進数の乗除算

- $3 \times 4 \rightarrow 3$ を4回 \_\_\_\_\_ と?  $12 \div 4 \rightarrow 12$ から4を何回 \_\_\_\_\_ ?  
 → 乗算と除算も \_\_\_\_\_ と \_\_\_\_\_ だけでOK! さらに \_\_\_\_\_ で効率化!
- シフト(演算): ビットを左右に \_\_\_\_\_

$3 \times 4$ 重み 8 4 2 1 $\begin{array}{r} 0011 \\ 0011 \\ 0011 \\ 0011 \\ \hline 11000 \end{array}$	$12 \div 4$ 重み 8 4 2 1 $\begin{array}{r} 1100 \\ 1100 \\ 1100 \\ \hline 0001100 \end{array}$	$3 \times 5$ $\begin{array}{r} (3:0011) \\ (5:0101) \\ 3 \times : 1100 \\ 3 \times : 0011 (+) \\ \hline \end{array}$	$15 \div 5$ $\begin{array}{r} (15:1111) \\ 1111 \\ -) 1010 : 5 \times \\ \hline 0101 \\ -) 0101 : 5 \times \\ \hline 0 \end{array}$
---	--	---	--

## ▶ 確認テスト

画面を止めて考えてみよう!

1. ビットを左右にずらすことを(① \_\_\_\_\_)という。  
 1ビット左にずらすと(② \_\_\_\_\_)をかけ算した数値になり、  
 1ビット右にずらすと(③ \_\_\_\_\_)で割り算した数値になる。
2. シフト演算で左に1がはみ出した場合を(④ \_\_\_\_\_)とい、右にはみ出したビットは割り算をしたときの(⑤ \_\_\_\_\_)を表している。

①補数 ②反転 ③1 ④反転 ⑤加算 (④⑤は順不同) 3. 1011 4. 1000



## ▶ ホワイトボード

▶ 実数

① \_\_\_\_\_ : 小数点の位置が固定 → 絶対値が大きい(細かい)値を \_\_\_\_\_

⑩進  $123.45_{(10)} \rightarrow 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$

②進  $101.01_{(2)} \rightarrow 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} =$  \_\_\_\_\_

② \_\_\_\_\_ : 小数点の位置が移動 → \_\_\_\_\_ してよく使われる

⑩進  $+ 6.02 \times 10^{23}$       ②進  $101.01 \times 2^0$

$10.101 \times 2^1$

$1.0101 \times 2^2$

$0.10101 \times 2^3$

$0.010101 \times 2^4$

⑩  $123.45$

⑩  $1.2345 \times 10^2$

$+ 0111111111 (+1023)$   
 $(+01111111 (+127))$

0 | 100 ..... 001 | 010100 ..... 00

1 (1)      11 (8)      52 (23)

+ + 0 - + 1      1023 (127) を足す      整数部分の 1 は省略

## ▶ 確認テスト

画面を止めて考えてみよう!

1. 実数の表現方法には小数点の位置を固定した(① \_\_\_\_\_)と、小数点の位置が移動する(② \_\_\_\_\_)がある。
2. 浮動小数点数の符号部では、正を(③ \_\_\_\_\_)で、負を(④ \_\_\_\_\_)で表現する。整数部分が1になるように仮数部と指数部を調整することを(⑤ \_\_\_\_\_)といい、仮数部では最上位の1を省略する。

①シフト ②2 ③2 ④あふれ ⑤あまり



▶ ホワイトボード

▶ 誤差

① \_\_\_\_\_ 誤差：下位の桁を切り捨て、切り上げ、四捨五入(\_\_\_\_\_)などで発生

$$0.2 = \underline{\quad} \times 2^{-1} + \underline{\quad} \times 2^{-2} + \underline{\quad} \times 2^{-3} + \underline{\quad} \times 2^{-4} + \underline{\quad} \times 2^{-5} + \underline{\quad} \times 2^{-6} + \underline{\quad} \times 2^{-7} + \underline{\quad} \times 2^{-8} + \dots$$

→ 0.001100110011....

▶ 桁が \_\_\_\_\_ データ型を使う!

② \_\_\_\_\_ 誤差：絶対値の小さい値を加減しても結果に反映されない

$$(1.0101 \times 2^1) + (1.1111 \times 2^{-6})$$

$$\begin{array}{r} 1.0101000000 \times 2^1 \\ +) 0.0000011111 \times 2^1 \\ \hline 1.0101 \quad \times 2^1 \end{array}$$

▶ 絶対値の \_\_\_\_\_ に計算する!

③ \_\_\_\_\_ 誤差：絶対値がほぼ等しい値を減算すると有効桁が減る

$$\begin{array}{r} \sqrt{101} = 0.10049875 \times 10^2 \\ -) \sqrt{100} = 0.10000000 \times 10^2 \\ \hline 0.00049875 \times 10^2 \\ 0.49875000 \times 10^{-1} \end{array}$$

▶ 減算しないように式を \_\_\_\_\_

$$\sqrt{101} - \sqrt{100} = \frac{(\sqrt{101} - \sqrt{100})(\sqrt{101} + \sqrt{100})}{\sqrt{101} + \sqrt{100}} = \frac{1}{\sqrt{101} + \sqrt{100}}$$

$$\sqrt{101} + \sqrt{100} = 0.20049875 \times 10^2$$

$$1 \div (0.20049875 \times 10^2) = 0.49875623 \times 10^{-1}$$

④ \_\_\_\_\_ 誤差：無限に続く計算を途中で打切ること発生

▶ 確認テスト

画面を止めて考えてみよう!

- 10進数の0.2は、2進数では0.00110011...の循環小数となる。この数値をコンピュータで有限の桁で表現するとき(① \_\_\_\_\_)誤差が発生する。
- 絶対値の小さい値を加減すると、計算結果に反映されない(② \_\_\_\_\_)誤差が発生する。絶対値のほぼ等しい値を減算すると、有効桁数が減る(③ \_\_\_\_\_)誤差が発生する。計算方法を工夫すると、誤差を減らすことができる。

①固定小数点数 ②浮動小数点数 ③0 ④1 ⑤正規化



## ▶ ホワイトボード

▶ 文字のデジタル表現

文字にはそれぞれ別々のビットパターンが割り当て( )られている

A ↔ 0100 0001 B ↔ 0100 0010

① 5つの文字コード

- 数字・英字・記号 (1)
- + 半角カナ (1), + 全角 (2)
- Windows UNIXなど
- 世界中の文字を統一, UTF-8, 16など

② 文字化けの理由

(1) 違う で表示

よろしく! → EUC → 1101 → UTF-8 → 1101 → 墮揚キリ.

(2) (特定の環境でしか表示されない文字, ①, I など) を使用

## ▶ 確認テスト

画面を止めて考えてみよう!

- 文字は0と1の組み合わせで表現されている。各文字をどのように表すかを取り決めたものを(① )という。(①)には、数字や英字、記号を1バイトで表現できる(② )コードや、世界各国の文字体系に対応させた(③ )などがある。
- 保存時と異なる文字コードで文字を表示すると、本来とは異なる文字が表示される。これを(④ )という。

①丸め ②情報落ち ③桁落ち



## ▶ ホワイトボード

▶ 音のデジタル表現

音(空気の振動)は ① \_\_\_\_\_ → ② \_\_\_\_\_ → ③ \_\_\_\_\_ でデジタル化される

・ \_\_\_\_\_ ( \_\_\_\_\_ ) 方式

電気信号

① 標本化( \_\_\_\_\_ )      ② 量子化      ③ 符号化( \_\_\_\_\_ )

精度

① 標本化 \_\_\_\_\_ : サンプルングの間隔?      ↓

② 標本化 \_\_\_\_\_ : 1秒間に何回? [Hz]      ↗

④ 量子化 \_\_\_\_\_ : 何段階に量子化? ↗

## ▶ 確認テスト

画面を止めて考えてみよう!

1. 電気信号に変換された音声をデジタル化するとき、まず音声の情報を一定の時間間隔に分割する。これを(① \_\_\_\_\_)といい、分割する間隔を(② \_\_\_\_\_)、1秒間に分割する回数を(③ \_\_\_\_\_)という。
2. (①)の後、電圧に対して一定間隔に分割して(④ \_\_\_\_\_)する。(④)した数値は0と1の組み合わせに置き換えて(⑤ \_\_\_\_\_)する。

①文字コード    ②ASCII    ③Unicode    ④文字化け



## ▶ ホワイトボード

▶ 色のデジタル表現

① ディスプレイ → \_\_\_\_\_ ( \_\_\_\_\_ )  
 - \_\_\_\_ (Red) · \_\_\_\_ (Green) · \_\_\_\_ (Blue)  
 - \_\_\_\_ + 赤・緑・青 → \_\_\_\_ : \_\_\_\_\_

② プリンタ → \_\_\_\_\_ ( \_\_\_\_\_ )  
 - \_\_\_\_ (Cyan) · \_\_\_\_ (Magenta) · \_\_\_\_ (Yellow)  
 - \_\_\_\_ + シ・マ・黄 → \_\_\_\_ : \_\_\_\_\_

④ 赤・緑・青の明るさをそれぞれ8ビットで表すと何種類の色を表現できる？  
 [参考] 色の三属性 (HSL)  
 - \_\_\_\_ (色合い: Hue)  
 - \_\_\_\_ (鮮やかさ: Saturation)  
 - \_\_\_\_ (明るさ: Lightness)

赤だけ → \_\_\_\_ = \_\_\_\_ 種類      0000 0000 (00)  
 赤 & 緑 & 青 → \_\_\_\_\_ )  
 = \_\_\_\_ 種類      1111 1111 (FF)

▶ \_\_\_\_ ビットで約 \_\_\_\_ 万色を表現 ( \_\_\_\_\_ ) ▶ 16進数6桁 #FFFFFF

## ▶ 確認テスト

### 画面を止めて考えてみよう!

1. 一般的に、ディスプレイは光の三原色とよばれる(① \_\_\_\_ )、(② \_\_\_\_ )、(③ \_\_\_\_ )を組み合わせるあらゆる色を表現する。この3色を混ぜると(④ \_\_\_\_ )に近づく。(①)、(②)、(③)の明るさを(⑤ \_\_\_\_ )ビットずつ割り当てると約1677万色を表現できる。これを(⑥ \_\_\_\_\_)という。
2. プリンタはシアン、マゼンダ、黄色の(⑦ \_\_\_\_ )の三原色を組み合わせる色を表現し、3色を混ぜると(⑧ \_\_\_\_ )に近づく。

- ①標本化 (サンプリング)    ②標本化周期    ③標本化周波数    ④量子化  
 ⑤符号化 (コード化)



## ▶ ホワイトボード

▶ 画像のデジタル表現

画像は ① \_\_\_\_\_ → ② \_\_\_\_\_ → ③ \_\_\_\_\_ でデジタル化される

① 標本化：画像を等間隔のマス目( \_\_\_\_\_ / \_\_\_\_\_ )に区切る

- \_\_\_\_\_ : デジタル画像の細かさ

・ディスプレイ → 横縦の \_\_\_\_\_ (1920×1080)

・プリンタ → 1インチあたりの \_\_\_\_\_ (300dpi)

② 量子化：色を数値で表現する

- \_\_\_\_\_ : 色の濃淡を何段階で表現するか

・ \_\_\_\_\_ → 白と黒のみ (1)

・ \_\_\_\_\_ → 白～黒の256段階 (8)

・ \_\_\_\_\_ → RGB各256段階 (24)

③ 符号化：量子化した数値を0と1(2進法)に変換

④ [ \_\_\_\_\_ ソフトウェア：画像を \_\_\_\_\_ で表現 → \_\_\_\_\_ 形式

\_\_\_\_\_ ソフトウェア：画像を \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_ などで表現 → \_\_\_\_\_ 形式

## ▶ 確認テスト

## 画面を止めて考えてみよう!

1. 画像のデジタル化では、まず画像を等間隔に読み取って(① \_\_\_\_\_)する。この細かく分けた最小単位を(② \_\_\_\_\_)といい、(②)の細かさのことを(③ \_\_\_\_\_)という。その後は(②)ごとに色の濃淡を数値で表現して(④ \_\_\_\_\_)する。最後に数値を0と1に変換して(⑤ \_\_\_\_\_)する。
2. 画像を点の集まりで表現する方法を(⑥ \_\_\_\_\_)形式、座標等の情報で表現する方法を(⑦ \_\_\_\_\_)形式という。

①赤 ②緑 ③青 (①②③は順不同) ④白 ⑤8 ⑥フルカラー ⑦色 ⑧黒



## ▶ ホワイトボード

▶ 動画のデジタル化

動画は \_\_\_\_\_ を連続的に表示したもの ( \_\_\_\_\_ で動いて見える )

① \_\_\_\_\_ : 動画を構成する静止画像  
\_\_\_\_\_ : 1秒あたりに表示される  
フレームの数 [ \_\_\_\_\_ ]

② フレームレート (大) → 動画  
→ データ量

③ 1920 × 1080 ピクセル・24ビットフルカラーの  
フレーム画像からなる 30 fps・1分間の  
動画のデータ量は?

## ▶ 確認テスト

画面を止めて考えてみよう!

1. 動画は ( ① \_\_\_\_\_ ) を連続的に表示したもので、見た画像が網膜に残る ( ② \_\_\_\_\_ ) 現象によって動いているように見えている。
2. 動画を構成する1枚ごとの静止画像を ( ③ \_\_\_\_\_ ) という。1秒あたりに表示される(③)の数を ( ④ \_\_\_\_\_ ) といい、単位を ( ⑤ \_\_\_\_\_ ) と表す。(④)が大きいほど滑らかな動画になるが、データ量は大きくなる。

①標本化 ②画素 (ピクセル) ③解像度 ④量子化 ⑤符号化 ⑥ラスタ ⑦ベクタ



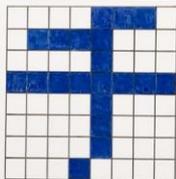
## ▶ ホワイトボード

▶ データの圧縮

① \_\_\_\_\_ : データの内容を保って  
データ量を減らす処理 ↔ \_\_\_\_\_  
→ 😊 \_\_\_\_\_ . \_\_\_\_\_  
圧縮率 [%] = \_\_\_\_\_ × 100

② \_\_\_\_\_ : 圧縮前に戻せる  
→ 文書データ、プログラム など  
\_\_\_\_\_ : 圧縮前に戻せない  
→ 画像、音声、動画 など

③ \_\_\_\_\_ : データの連続回数に変換



④ 画像 : BMP、GIF、PNG、JPEG  
音声 : WAVE、MP3、AAC、WMA  
動画 : AVI、MOV、MPEG など

□ 白を0、青を1 →  
⑤ データの連続回数を並べると  
4, 3, 2, 4, 7, 1, 3, 8, 4, 1,  
7, 1, 7, 1, 6, 1, 4 [17個]  
→  
圧縮率 =

## ▶ 確認テスト

画面を止めて考えてみよう!

1. データの内容を保ったままデータ量を小さくする処理を  
(① \_\_\_\_\_)といい、(①)したデータを元に戻すことを(②  
\_\_\_\_\_)という。(①)前と(②)後が同じになる(③ \_\_\_\_\_  
\_\_\_\_\_)と多少の変更を許容する(④ \_\_\_\_\_)がある。
2. 同じデータを連続回数の数値に置き換えて表現する方式を  
(⑤ \_\_\_\_\_)圧縮という。100ビットのデータが  
70ビットに圧縮されたとき、圧縮率は(⑥ \_\_\_\_\_)%となる。

①静止画像 ②残像 ③フレーム ④フレームレート ⑤fps



## ▶ ホワイトボード

▶ コミュニケーションとメディアの変化

- \_\_\_\_\_ ( \_\_\_\_\_ ) : 情報をやりとりすること
- \_\_\_\_\_ : 情報をやりとりする方法や媒体. 不特定多数向け → \_\_\_\_\_

	15世紀	18世紀	19世紀	20世紀	1969	1991
技術						
個人						
マス						

- \_\_\_\_\_ : 多数の人々が相互に情報発信するコミュニケーションサービス

## ▶ 確認テスト

画面を止めて考えてみよう!

1. 情報をやりとりすることを(① \_\_\_\_\_)といい、その方法や媒体を(② \_\_\_\_\_)という。
2. 情報のやりとりは口頭から始まり、15世紀には活版印刷の発明によって(③ \_\_\_\_\_)や(④ \_\_\_\_\_)などの印刷物が普及した。19世紀には電気通信による(⑤ \_\_\_\_\_)で遠くの人と瞬時にやりとりできるようになった。20世紀後半には(⑥ \_\_\_\_\_)により、双方向の情報伝達が可能になった。

①圧縮 ②展開(解凍) ③可逆圧縮 ④非可逆圧縮 ⑤ランレングス ⑥70



## ▶ ホワイトボード

▶ コミュニケーションとネットリテラシ

① コミュニケーションの形

	同期型	非同期型
1対1	会話 電話 ビデオ通話	LINE メール 手紙
1対多	授業 プレゼン ライブ配信	SNS HP YouTube
	○ 反応が分かる △ 時間・場所の制約	○ 便利、蓄積しやすい △ 勘違い、消せない

② ネットコミュニケーションの特徴と注意点

① \_\_\_\_\_ : 名前などの情報を隠せる

② \_\_\_\_\_ が保証されていない  
→ 情報を \_\_\_\_\_ にしない!

③ \_\_\_\_\_ : 誰にでも見られる可能性

④ \_\_\_\_\_ : 発信した情報は消せない  
→ \_\_\_\_\_ で言わないことは言わない!

⑤ \_\_\_\_\_ : アクセス情報はすべて記録  
→ ネットは \_\_\_\_\_ ではない!

## ▶ 確認テスト

画面を止めて考えてみよう!

1. コミュニケーションには、相手からすぐに反応がある(① \_\_\_\_\_ )と、相手がいつ受信したか分からない(② \_\_\_\_\_ )がある。その時々で適切な方法を選ぶことが大切である。
2. ネットには役立つ情報があるが、名前を隠せる(③ \_\_\_\_\_ )を考慮しなければならない。また、発信者の不注意で情報が誤っている可能性もあり、(④ \_\_\_\_\_ )が保証されていない。情報を鵜呑みにしないことが大切である。

①通信 ②メディア ③書籍 ④新聞 (③④は順不同) ⑤電話 ⑥インターネット



## ▶ ホワイトボード

▶ 情報デザイン

情報を分かりやすく伝えるための工夫や表現方法

① 余分な情報を除いて要点をシンプルに表現  
 (案内誘導板)、

② 視覚的に分かりやすく表現  
 図、表、グラフ、  
 (路線図、ハザードマップ)

③ 情報を整理して分かりやすく表現

## ▶ 確認テスト

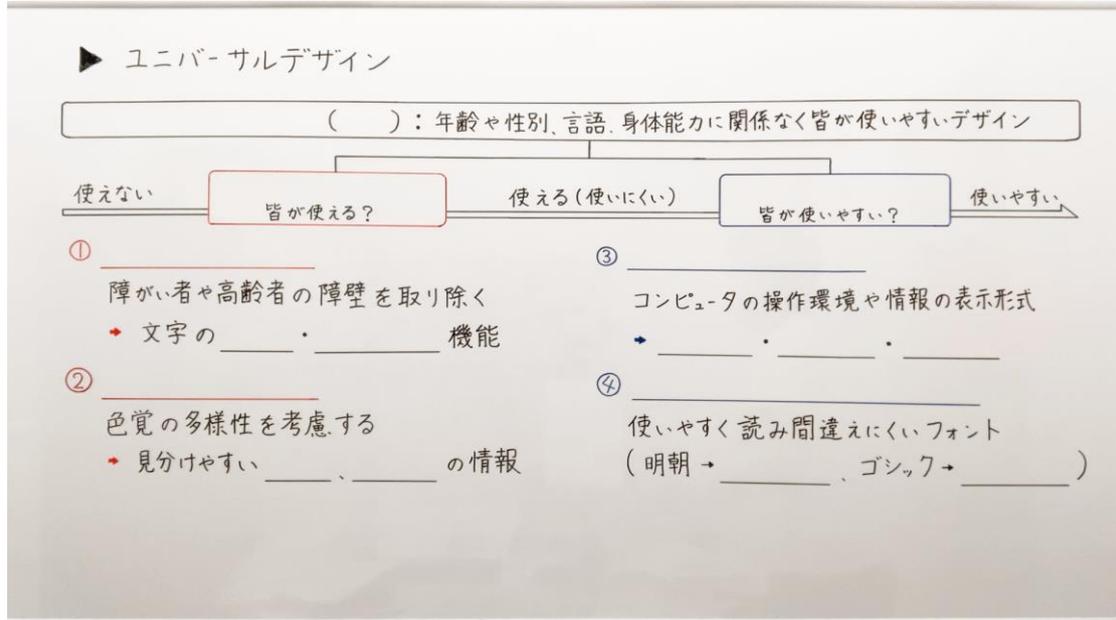
画面を止めて考えてみよう!

1. 情報を分かりやすく伝えるための工夫や表現方法を(① \_\_\_\_\_)といい、メディアの多様化や情報量の増加が進む現代社会ではとても重要なものとなっている。
2. (①)の手法には、余分な情報を除いて要点をシンプルに表現する(② \_\_\_\_\_)、情報を視覚的に分かりやすく表現する(③ \_\_\_\_\_)、情報を整理して分かりやすく表現する(④ \_\_\_\_\_)などがある。

①同期型 ②非同期型 ③匿名性 ④信憑性



## ▶ ホワイトボード

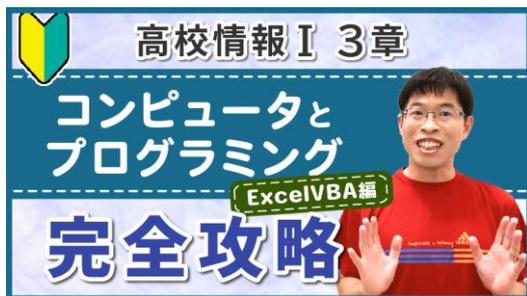


## ▶ 確認テスト

画面を止めて考えてみよう!

- 年齢や性別、言語、身体能力などに関係なく、皆にとって使いやすいものを設計することを(① \_\_\_\_\_)という。皆が使えるかの尺度を(② \_\_\_\_\_)、使いやすいかの尺度を(③ \_\_\_\_\_)という。
- コンピュータの操作環境や情報の表示形式などを(④ \_\_\_\_\_)といい、情報にアクセスしやすいことやストレスなく操作できることなどが優れた(④)につながる。

①情報デザイン ②抽象化 ③可視化 ④構造化



【3章】 アルゴリズムやプログラミングを徹底解説！

 <https://youtu.be/sHLLw4ICpPI>

動画時間：5:21:34

学習項目：29



### 3章のゴール

- コンピュータを構成している装置や回路を理解しよう
- アルゴリズムとは何かや表現方法を理解して、プログラムで実装したい処理の流れを考えられるようになる
- 表計算マクロ言語で変数や配列、関数などを用いたプログラムを記述して実行できるようになる
- モデル化やシミュレーションの考え方を理解して、プログラムを用いて実際にシミュレーションできるようになる

### ▶ キーワード

- |                                       |   |                                    |
|---------------------------------------|---|------------------------------------|
| <input type="checkbox"/> 入力装置         | <input type="checkbox"/> AND・OR・NOT 演算  | <input type="checkbox"/> 変数        |
| <input type="checkbox"/> 記憶装置         | <input type="checkbox"/> 半加算・全加算回路      | <input type="checkbox"/> 配列        |
| <input type="checkbox"/> 主記憶装置・メインメモリ | <input type="checkbox"/> 基本ソフトウェア       | <input type="checkbox"/> プログラム     |
| <input type="checkbox"/> 補助記憶装置       | <input type="checkbox"/> OS             | <input type="checkbox"/> 表計算マクロ言語  |
| <input type="checkbox"/> 演算装置         | <input type="checkbox"/> アプリケーションソフトウェア | <input type="checkbox"/> Excel VBA |
| <input type="checkbox"/> 出力装置         | <input type="checkbox"/> アルゴリズム         | <input type="checkbox"/> 演算子       |
| <input type="checkbox"/> 制御装置         | <input type="checkbox"/> フローチャート        | <input type="checkbox"/> 関数        |
| <input type="checkbox"/> CPU・中央処理装置   | <input type="checkbox"/> アクティビティ図       | <input type="checkbox"/> API       |
| <input type="checkbox"/> インタフェース      | <input type="checkbox"/> 状態遷移図          | <input type="checkbox"/> モデル化      |
| <input type="checkbox"/> クロック周波数      | <input type="checkbox"/> 順次・分岐・反復構造     | <input type="checkbox"/> シミュレーション  |
| <input type="checkbox"/> 論理演算         |   |                                    |



## ▶ ホワイトボード

▶ コンピュータの構成

(各装置を制御)

(データの演算)

(データを入力)

(データを一時的に保存)

(データを長期的に保存)

(データを出力)

- ・ \_\_\_\_\_ : コンピュータを構成する装置
- ・ \_\_\_\_\_ : 装置同士を接続する規格. USB や Wi-Fi, Bluetooth など.

## ▶ 確認テスト

画面を止めて考えてみよう!

① コンピュータを構成する装置には、外部からデータを取り込む(① \_\_\_\_\_)装置、データを記憶する記憶装置、データを演算する(② \_\_\_\_\_)装置、外部へデータを出力する(③ \_\_\_\_\_)装置、これらの装置を制御する(④ \_\_\_\_\_)装置がある。記憶装置にはデータを一時的に保存する(⑤ \_\_\_\_\_)装置と、長期的に保存できる(⑥ \_\_\_\_\_)装置がある。このような物理的な装置を(⑦ \_\_\_\_\_)という。

- ①ユニバーサルデザイン ②アクセシビリティ ③ユーザビリティ  
④ユーザインタフェース



## ▶ ホワイトボード

▶ CPUとメモリの仕組み

制御装置

プログラムカウンタ 0  
次の命令のアドレス

命令レジスタ  
命令を保存

命令解読器      クロックジェネレータ

---

演算装置

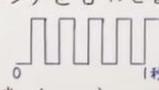
汎用レジスタ  
A B C

算術論理演算装置

主記憶装置

命令・データ	アドレス
READ A. 100	0
READ B. 101	1
ADD A. B	2
WRITE 102. C	3
STOP	4
...	...
10	100
20	101
...	102

- CPU内にある記憶領域
- 各装置の動作タイミングを合わせる周期的な信号



1秒間の信号の回数( )

- 処理効率を上げる技術
- CPUの \_\_\_\_\_ : 32 → 64
- \_\_\_\_\_ を上げる
- ③ \_\_\_\_\_

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① CPUが命令を実行するときは、まず(① \_\_\_\_\_ )で指定された主記憶装置のアドレスを参照して、命令を(② \_\_\_\_\_ )に取り出す。その命令を(③ \_\_\_\_\_ )で解読して実行する。
- ② 各装置の動作タイミングを合わせるための周期的な信号を(④ \_\_\_\_\_ )という。1秒間あたりの(④)の発生回数を(⑤ \_\_\_\_\_ )といい、単位を(⑥ \_\_\_\_\_ )で表す。

①入力 ②演算 ③出力 ④制御 ⑤主記憶 ⑥補助記憶 ⑦ハードウェア



## ▶ ホワイトボード

▶ 演算の仕組み

演算装置は \_\_\_\_\_ (真と偽のAND、ORなど)や \_\_\_\_\_ (0と1の+、-など)ができる

① 基本的な論理回路

(論理積) (論理和) (否定)

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

A	Y
0	1
1	0

A B → Y (AND)      A B → Y (OR)      A → Y (NOT)

② \_\_\_\_\_ : 2進数1桁の加算回路

A	B	C	S
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

③ \_\_\_\_\_ : 桁上りを考慮した加算回路

A B C' → C S (Full Adder)

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 基本的な論理回路として(① \_\_\_\_\_)(論理積)回路、(② \_\_\_\_\_)(論理和)回路、(③ \_\_\_\_\_)(否定)回路がある。この3つの回路を組み合わせることで様々な演算を実現できる。
- ② 2進数1桁の加算回路を(④ \_\_\_\_\_)といい、基本的な論理回路の組み合わせで作れる。さらに(④)を2つ組み合わせると、桁上りを考慮した(⑤ \_\_\_\_\_)を作ることができる。

- ①プログラムカウンタ ②命令レジスタ ③命令解読器 ④クロック信号  
⑤クロック周波数 ⑥Hz



## ▶ ホワイトボード

▶ ソフトウェアの種類

- OSの管理機能
  - ① \_\_\_\_\_  
複数の処理(タスク)に対してCPUを適切に割り当て
  - ② \_\_\_\_\_  
主記憶装置を効果的に活用
  - ③ \_\_\_\_\_  
フォルダ(ディレクトリ)によってファイルを階層的に管理

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① ハードディスクやディスプレイのようにコンピュータを構成する物理的な装置を(① \_\_\_\_\_)といい、(①)上で動くプログラムなどを(② \_\_\_\_\_)という。
- ② ソフトウェアには特定の作業に使う(③ \_\_\_\_\_)ソフトウェアや、ハードウェアを有効活用するための(④ \_\_\_\_\_)ソフトウェアがある。Windowsなどの(⑤ \_\_\_\_\_)(OS)も(④)に含まれる。

①AND ②OR ③NOT ④半加算回路 ⑤全加算回路



## ▶ ホワイトボード

▶ アルゴリズムとは

① \_\_\_\_\_ ( \_\_\_\_\_ )

② \_\_\_\_\_

③ \_\_\_\_\_

▶ アルゴリズムは \_\_\_\_\_ 考えられる. \_\_\_\_\_ & \_\_\_\_\_ & \_\_\_\_\_ が大切!

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 処理や計算の手順を(① \_\_\_\_\_)といい、(①)をコンピュータが実行できるように記述したものを(② \_\_\_\_\_)という。
- ② アルゴリズムを分かりやすく表現する方法として、流れ図とも呼ばれる(③ \_\_\_\_\_)や、並行した処理を表現しやすい(④ \_\_\_\_\_)、状態の移り変わりを表現する(⑤ \_\_\_\_\_)がある。

- ①ハードウェア ②ソフトウェア ③アプリケーション ④基本
- ⑤オペレーティングシステム



## ▶ ホワイトボード

▶ フローチャートの基本

- \_\_\_\_\_ ( \_\_\_\_\_ ): 処理の手順(アルゴリズム)を処理記号などで図にしたもの  
[合計金額を求める]

--	--	--	--

- 変化する値は \_\_\_\_\_ で表す. 変数に値を入れる( \_\_\_\_\_ )ときは「 \_\_\_\_\_ 」と書く
- 処理の流れを追うことを \_\_\_\_\_ という. \_\_\_\_\_ を書くのがオススメ!

## ▶ 確認クイズ: ペンを4本・ノートを2冊にするとどう変わる?

フローチャートの基本 フローチャートとは

[合計金額を求める]

端子記号

処理記号

- 変化する値は \_\_\_\_\_ で表す. 変数に値を入れる( \_\_\_\_\_ )

- ①アルゴリズム ②プログラム ③フローチャート ④アクティビティ図 ⑤状態遷移図



▶ ホワイトボード

▶ 構造化プログラミング

右の3つの構造を組み合わると  
アルゴリズムが分かりやすくなる

▶ 分岐構造

ペンノート合計

① 処理  
処理  
処理

② 条件  
Yes 処理1  
No 処理2

③ 条件  
処理

開始  
100 × 5 → ペン  
150 × 4 → ノート  
ペン + ノート → 合計  
合計 ≥ 1000  
Yes と同じ  
No  
合計 × 0.9 → 合計  
終了

開始  
x:y >  
x:z ≤  
y:z ≤  
x → A  
z → A  
y → A  
終了

Q1 x = 1, y = 2, z = 3  
A = ?

Q2 x = 2, y = 3, z = 1  
A = ?

Q3 x = 3, y = 1, z = 2  
A = ?

▶ 確認クイズ : Q1~Q3 A の値はいくつになるか

分岐構造 確認クイズ

開始  
x:y >  
x:z ≤  
y:z ≤  
x → A  
z → A  
y → A  
終了

Q1 x = 1, y = 2, z = 3  
A = ?

Q2 x = 2, y = 3, z = 1  
A = ?

Q3 x = 3, y = 1, z = 2  
A = ?

(前ページの回答は動画参照)



▶ ホワイトボード

▶ 反復構造

構造化プログラミングで使う3構造の1つ. \_\_\_\_\_ や \_\_\_\_\_ ともいう.

<pre> 開始       O → A     → n       A + n → A       n + 1 → n       n : 3       &gt;       終了           </pre>	<p><u>    n  A    </u></p>	<pre> 開始       O → A     → n       加算ループ       A + n → A       n + 1 → n       n &gt; 3 まで   加算ループ       終了           </pre>	<p><u>    n  A    </u></p>	<p>Q トレースしてみよう!</p> <p><u>    α  b  γ    </u></p> <pre> 開始       2 → a       4 → b       1 → γ       ループ       γ × a → γ       b - 1 → b       b = 0 まで   ループ       終了           </pre>
---	----------------------------	--	----------------------------	---

▶ 確認クイズ：トレースしてみよう！

分岐構造 確認クイズ

<pre> 開始       → A       → n       算ループ       - n → A       - 1 → n       &gt; 3 まで   ループ       了           </pre>	<p><u>    n  A    </u></p> <p>0</p> <p>1 1</p> <p>2 3</p> <p>3 6</p> <p>4</p>	<p>Q トレースしてみよう!</p> <p><u>    α  b  γ    </u></p> <pre> 開始       2 → a       4 → b       1 → γ       ループ       γ × a → γ       b - 1 → b       b = 0 まで   ループ       終了           </pre>
--	---	---

ループ端  
終了条件 (脱出条件)



## ▶ ホワイトボード

▶ 配列

- \_\_\_\_\_ : 複数のデータをまとめて管理するもの

```

開始
  |
  0 → S
  1 → i
  |
  合計ループ
  i > 5まで
  |
  S + score(i) → S
  |
  i + 1 → i
  |
  合計ループ
  |
  終了
          
```

score(1)	80
:(2)	65
:(3)	85
:(4)	70
:(5)	90

Q Mは何?

```

開始
  |
  score(1) → M
  2 → i
  |
  ループ
  i < 5まで
  |
  判定
  score(i) : M
  |
  >
  |
  score(i) → M
  |
  i + 1 → i
  |
  ループ
  |
  終了
          
```

Mは \_\_\_\_\_

## ▶ 確認クイズ : Mの値はいくつになるか

とめて管理するもの

**要素**

score(1)	80
:(2)	65
:(3)	85
:(4)	70
:(5)	90

**添字 (要素の番号)**

分岐構造 確認クイズ

Q Mは何?

```

開始
  |
  score(1) → M
  2 → i
  |
  ループ
  i > 5まで
  |
  判定
  score(i) : M
  |
  >
  |
  score(i) → M
  |
  i + 1 → i
  |
  ループ
  |
  終了
          
```

Mは \_\_\_\_\_

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ 線形探索法

- \_\_\_\_\_ : 多くのデータから特定のデータを探す  
 ↳ \_\_\_\_\_ (最初から順々に探す)、\_\_\_\_\_ など

no	name
18	菅原
15	桜井
21	田原
27	中川
19	鈴木
08	小林
25	寺岡

## ▶ 確認クイズ：トレースしてみよう！（no(8)の役割は？）

線形探索法 確認クイズ

no	name
18	菅原
15	桜井
21	田原
27	中川
19	鈴木
08	小林
25	寺岡
21	8

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ 二分探索法

探索範囲を \_\_\_\_\_ にすることを繰り返す探索方法  
 \_\_\_\_\_ だけデータを \_\_\_\_\_ しておく必要有

name	no	
小林	1	08
桜井	2	15
菅原	3	18
鈴木	4	19
田原	5	21
寺岡	6	25
中川	7	27

探索キー 21      L    i    H

Q 05

```

graph TD
    Start([開始]) --> Init["1 → L  
7 → H"]
    Init --> CalcMid["[(L+H)÷2] → i"]
    CalcMid --> Compare["キ: no(i)"]
    Compare --> Less["<"]
    Compare --> Greater[">"]
    Compare --> Equal["="]
    Less --> UpdateL["i-1 → H"]
    UpdateL --> CalcMid
    Greater --> UpdateR["i+1 → L"]
    UpdateR --> CalcMid
    Equal --> Found["なし"]
    Found --> End([終了])
    Found --> Name["name(i)"]
    Name --> End
  
```

## ▶ 確認クイズ：トレースしてみよう！（探索キー：05）

法

半分 にすることを繰り返す探索方法  
 データを 整列 しておく必要有

二分探索法 確認クイズ

[x]: xを超えない最大の整数  
 [4] = 4, [5.5] = 5

no	探索キー	L	i	H
1	21	1	4	7
2		5	6	
3			5	5
4				
5				
6	Q 05	1		7
7				

no: 08 L, 15, 18, 19, 21, 25, 27 H

```

graph TD
    Start([開始]) --> Init["1 → L  
7 → H"]
    Init --> CalcMid["[(L+H)÷2] → i"]
    CalcMid --> Compare["キ: no(i)"]
    Compare --> Less["<"]
    Compare --> Greater[">"]
    Compare --> Equal["="]
    Less --> UpdateL["i-1 → H"]
    UpdateL --> CalcMid
    Greater --> UpdateR["i+1 → L"]
    UpdateR --> CalcMid
    Equal --> Found["なし"]
    Found --> End([終了])
    Found --> Name["name(i)"]
    Name --> End
  
```

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ 基本交換法 (バブルソート)

- \_\_\_\_\_ : データを順番に並べる
- 基本交換法 (バブルソート)  
\_\_\_\_\_ のデータを比べて逆順のときに入れかえる

A	5	7	1	9	3
	5			9	3
	5	1	7		
			7	3	9
	1	5			9
				7	9

```

graph TD
    Start([開始]) --> I5[5 → I]
    I5 --> Loop1[全体ループ  
I < 2 まで]
    Loop1 --> J[1 → J]
    J --> Loop2[交換ループ  
J > I-1 まで]
    Loop2 --> Compare{A(J):A(J+1)}
    Compare -- ≤ --> Jplus[J+1 → J]
    Compare -- > --> Swap[A(J) → T  
A(J+1) → A(J)  
T → A(J+1)]
    Swap --> Loop2
    Jplus --> Loop1
    Loop2 --> Loop3[交換ループ  
I-1 → I]
    Loop3 --> Loop1
    Loop1 --> End([終了])
  
```

I	J	A
		5 7 1 9 3

## ▶ 確認クイズ: トレースしてみよう! (I=3,2,1)

基本交換法 確認クイズ

番に並べる  
-ト)  
順のときに

3
3
9
9
9
9

```

graph TD
    Start([開始]) --> I5[5 → I]
    I5 --> Loop1[全体ループ  
I < 2 まで]
    Loop1 --> J[1 → J]
    J --> Loop2[交換ループ  
J > I-1 まで]
    Loop2 --> Compare{A(J):A(J+1)}
    Compare -- ≤ --> Jplus[J+1 → J]
    Compare -- > --> Swap[A(J) → T  
A(J+1) → A(J)  
T → A(J+1)]
    Swap --> Loop2
    Jplus --> Loop1
    Loop2 --> Loop3[交換ループ  
I-1 → I]
    Loop3 --> Loop1
    Loop1 --> End([終了])
  
```

I	J	A
		5 7 1 9 3
5	1	
	2	1 7
	3	
	4	3 9
	5	5 1 7 3 9
4	1	1 5
	2	
	3	3 7
	4	1 5 3 7 9
3		

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ 基本選択法 (選択ソート)

データを選擇して  
右端に置く

A	5	7	1	9	3
	5	7	1		
	5		1	9	
		3		7	9
			5	7	9

開始  
5 → I  
全体ループ I < 2 まで  
I → MAX  
I - 1 → J  
選択ループ J < I まで  
A(J):A(MAX) >  
J → MAX  
J - 1 → J

選択ループ  
A(I) → T  
A(MAX) → A(I)  
T → A(MAX)  
I - 1 → I  
全体ループ  
終了

I	J	M	A
		5	7   9 3

## ▶ 確認クイズ: トレースしてみよう! (I=4,3,2,1)

ソート) 選択して

基本選択法 確認クイズ

I	J	M	A
		5	7   9 3
5	4	5	
	3	4	
	2		
	1		
	0		5 7   3 9
4			

開始  
5 → I  
全体ループ I < 2 まで  
I → MAX  
I - 1 → J  
選択ループ J < I まで  
A(J):A(MAX) >  
J → MAX  
J - 1 → J

選択ループ  
A(I) → T  
A(MAX) → A(I)  
T → A(MAX)  
I - 1 → I  
全体ループ  
終了

3  
9  
9  
9  
9

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ 基本挿入法 (挿入ソート)

\_\_\_\_\_ から整列しながら、途中に入るデータがあれば挿入する

A	5	7	1	9	3
	5	—	—	—	—
	5	7	—	—	—
	1	5	7	—	—
	1	5	7	9	—

I	J	A
5	7	1
7	1	9
1	9	3

```

graph TD
    Start([開始]) --> I2[2 → I]
    I2 --> Loop1[全体ループ  
I > 5 まで]
    Loop1 --> J1[I - 1 → J]
    J1 --> Loop2[挿入ループ  
J < 1 まで]
    Loop2 --> Cond{A(J):A(J+1)}
    Cond -->|≧| Shift[A(J) → T  
A(J+1) → A(J)  
T → A(J+1)]
    Shift --> J1
    Cond -->|>| J0[0 → J]
    J0 --> J1
    Loop2 --> Loop3[挿入ループ  
I + 1 → I]
    Loop3 --> Loop1
    Loop1 --> End([終了])
  
```

## ▶ 確認クイズ：トレースしてみよう！

途中に  
る

I	J	A
5	7	1
7	1	9
1	9	3

```

graph TD
    Start([開始]) --> I2[2 → I]
    I2 --> Loop1[全体ループ  
I > 5 まで]
    Loop1 --> J1[I - 1 → J]
    J1 --> Loop2[挿入ループ  
J < 1 まで]
    Loop2 --> Cond{A(J):A(J+1)}
    Cond -->|≧| Shift[A(J) → T  
A(J+1) → A(J)  
T → A(J+1)]
    Shift --> J1
    Cond -->|>| J0[0 → J]
    J0 --> J1
    Loop2 --> Loop3[挿入ループ  
I + 1 → I]
    Loop3 --> Loop1
    Loop1 --> End([終了])
  
```

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ プログラミング

① プログラミングの流れ

STEP 1 :  
どの? (Python, Java ...) どのような?

STEP 2 :  
[1] `sum = 10 + 20`  
`print (sum)` → [2] `01101...` → [3] `30`

STEP 3 :

② プログラミング言語の種類

`MOV, ADDA`  
`sum = a + b`  
まとめて翻訳: C, Java...  
1つずつ翻訳: Python, JavaScript...

<b>Python</b> シンプルで分かりやすい 人工知能やデータ解析など	<b>JavaScript</b> webブラウザ向けに開発 動きのあるwebページなど
<b>表計算マクロ言語</b> 表計算ソフト上で動く 計算などの処理を自動化	<b>Scratch</b> 学習用ビジュアル型言語 ブロックで直観的に開発

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① プログラミングは(① \_\_\_\_\_)、(② \_\_\_\_\_)、  
(③ \_\_\_\_\_)の手順で進めることが多い。(①)では使用するプログラミング(④ \_\_\_\_\_)を決めたり、プログラムの流れを(⑤ \_\_\_\_\_)などで表現したりする。
- ② プログラムは人にとって分かりやすいソースコードから、コンピュータが理解できる0と1の(⑥ \_\_\_\_\_)に変換して実行される。この変換を(⑦ \_\_\_\_\_)とよぶ。

(前ページの回答は動画参照)



## ▶ ホワイトボード

▶ 表計算マクロ言語とは

① ソフトウェアの \_\_\_\_\_ 機能を実現するプログラミング \_\_\_\_\_

Excel  
Google  
スプレッドシートなど



大きな処理(複数の処理をまとめたもの)を \_\_\_\_\_  
[例] データチェック → 集計 → 書類作成



② \_\_\_\_\_ (Visual Basic for Applications) Word, Excel, PowerPoint など

\_\_\_\_\_ 年にMicrosoft社が公開したOfficeシリーズで動くプログラミング言語

→ 作業の \_\_\_\_\_、アプリ間の \_\_\_\_\_、OSの \_\_\_\_\_

特徴 ① 特別な \_\_\_\_\_ が不要!

② プログラムが \_\_\_\_\_ かつ \_\_\_\_\_ !

⚠ \_\_\_\_\_ に要注意! → \_\_\_\_\_ できるマクロだけ実行しよう



ココ大切!

(確認テストやプログラムはありません)

- ①設計
- ②コーディング
- ③テスト
- ④言語
- ⑤フローチャート
- ⑥機械語
- ⑦翻訳





## ▶ ホワイトボード

▶ VBA の基本

① プログラムの構造

```
Sub プロシージャ名()
  (処理)
End Sub
```

モジュール  
プロシージャ  
...

② 基本的なルール

- 処理は \_\_\_\_\_ で区切る
- 処理のまとまりを \_\_\_\_\_ ( \_\_\_\_\_ ) で指定する
- 文字列は \_\_\_\_\_ で囲む
- コメント ( \_\_\_\_\_ 以降) でメモ書きを入れる

POINT!

③ セルの指定方法

	A	B
1		
2		
3		

Cells(行, 列).Value  
Range("セル範囲").Value  
Cells(3, 2).Value = 10

④ 実行ボタンの作成

[開発]タブ - [挿入] - [フォームコントロール]

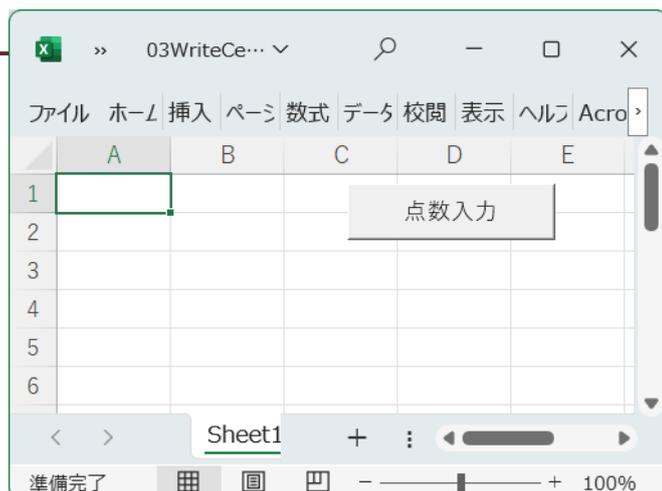
'セルに点数を入力するプログラム

```
Sub WriteCell()
  Cells(1, 1).Value = "英語"
  Range("B1").Value = 90
End Sub
```

## ▶ プログラム

### 03WriteCell.xlsm

```
1 'セルに点数を入力するプログラム
2 Sub WriteCell()
3   Cells(1, 1).Value = "英語"
4   Range("B1").Value = 90
5 End Sub
```





## ▶ ホワイトボード

▶ 変数

① 変数はデータの \_\_\_\_\_

② 変数名のルール 

- ・ 英数字、ひらがな、カタカナ、漢字、\_ (アンダースコア) を使える
- ・ 大文字・小文字は区別 \_\_\_\_\_
- name, student\_no
- ・ |文字目に \_\_\_\_\_ は使えない
- ・ キーワード( \_\_\_\_\_ )は使えない
- ✗ 1st\_number, sub

③ 変数の使い方

① \_\_\_\_\_ する : 変数を作る  
Dim 変数名(, 変数名, ...)  
Dim age

② \_\_\_\_\_ する : 値を入れる  
変数名 = 値  
age = 16

③ \_\_\_\_\_ する : 値を見る  
変数名 (→ 値に置きかわる)  
Range("B1").Value = age 

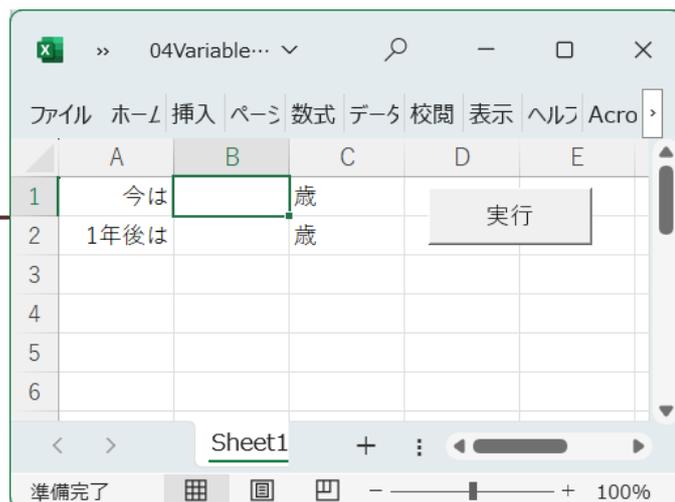
## ▶ プログラム

### 04Variable.xlsm

```

1 Sub Variable()
2   Dim age
3   age = 16
4   Range("B1").Value = age
5
6   age = age + 1
7   Range("B2").Value = age
8 End Sub

```





## ▶ ホワイトボード

▶ データ型

① データ型：データの \_\_\_\_\_

- 整数 (10, 365)
- 小数 (3.1415...)
- 文字列 ("VBA")
- 真偽値 (True, False)

② 変数の宣言

```
Dim 変数名 As データ型
Dim age As Byte
age = 16
age = 256 →
```

整数	Byte	8ビット	0 ~ 255
	Integer	16	-32,768 ~ 32,767
	Long	32	-2,147,483,648 ~ 2,147,483,647
小数	Single	32	単精度 浮動小数点数
	Double	64	倍精度 =
文字列	String	80 + 文字列長	~ 約 65,400 文字
真偽値	Boolean	16	True, False のみ
日付	Date	64	100年1月1日 ~ 9999年12月31日
バリエーション	Variant	128	[数値] Doubleと同じ
		170 + 文字列長	[文字列] Stringと同じ

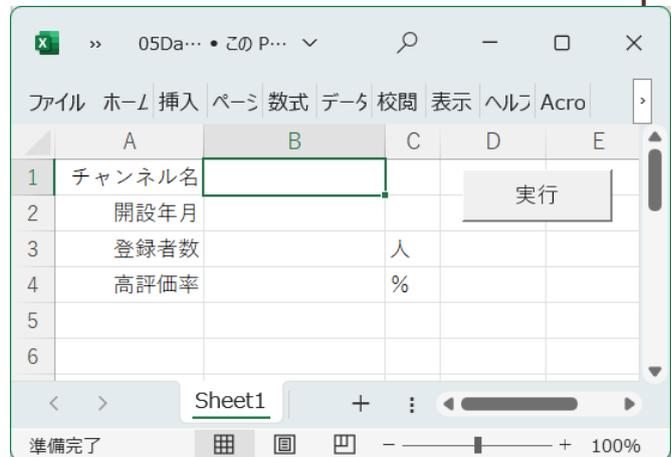
## ▶ プログラム

05DataType.xlsm

```

1 Sub DataType()
2   Dim name As String
3   Dim year As Integer, month As Integer
4   Dim count As Long
5   Dim rate As Double
6
7   name = "せかちゃん"
8   year = 2021
9   month = 9
10  count = 20000
11  rate = 98.4
12
13  Range("B1").Value = name
14  Range("B2").Value = year & "年" & month & "月"
15  Range("B3").Value = count
16  Range("B4").Value = rate
17 End Sub

```





## ▶ ホワイトボード

### ▶ 配列

① \_\_\_\_\_ : 複数の変数をまとめて管理

② 配列の使い方

① \_\_\_\_\_ する

Dim 配列名(要素数-1) As データ型  
: (最小値 To 最大値) :

Dim score(2) As Long

Dim score(0 To 2) As Long

② \_\_\_\_\_ する

配列名(添字) = 値  
score(0) = 80

③ \_\_\_\_\_ する

配列名(添字)  
score(0)

score

80	100	90
0	1	2



③ 便利な関数

処理をまとめたもの

■ Array(要素, 要素, ...): 配列の作成

Dim name As Variant

name = Array("菅原", "梅井", "田原")

■ LBound(配列名): 最小の要素番号

UBound(配列名): 最大

LBound(name) →

UBound(name) →



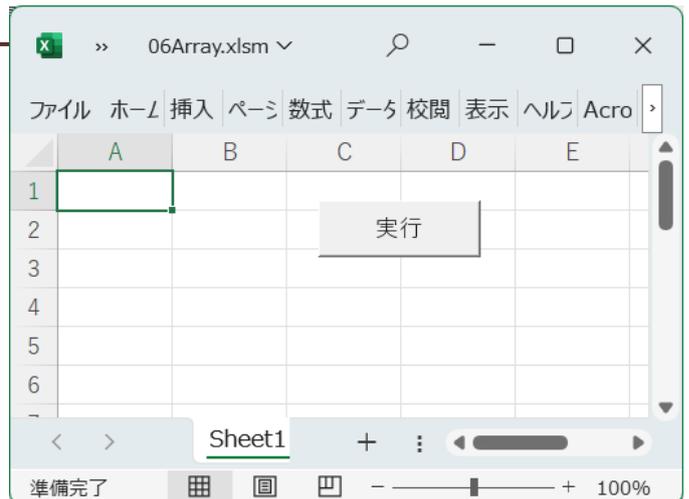
## ▶ プログラム

### O6Array.xlsm

```

1 Sub ScoreArray()
2   Dim score(2) As Long
3   score(0) = 80
4   score(1) = 100
5   score(2) = 90
6
7   Dim name As Variant
8   name = Array("菅原", "桜井", "田原")
9
10  Range("A1").Value = name(0)
11  Range("A2").Value = name(1)
12  Range("A3").Value = name(2)
13  Range("B1").Value = score(0)
14  Range("B2").Value = score(1)
15  Range("B3").Value = score(2)
16
17  Range("A4").Value = "受験者数"
18  Range("B4").Value = (UBound(name) + 1) & "名"
19 End Sub

```





## ▶ ホワイトボード

▶ 演算子

①	^	累乗
	* /	× ÷
	\ (¥) Mod	商 余り
	+ -	+ -
②	&	連結
③	< > <= >=	大小比較
	= <>	等しいか
④	Not	～ない
	And	かつ
	Or	または

① 算術演算子  
 $7^2 \rightarrow \underline{\quad}$      $7/2 \rightarrow \underline{\quad}$   
 $7 \setminus 2 \rightarrow \underline{\quad}$      $7 \text{ Mod } 2 \rightarrow \underline{\quad}$

② 文字列連結演算子  
 $7 \& 2 \rightarrow \underline{\quad}$

③ 比較演算子  
 $a \geq 0 \rightarrow a$ が0以上なら  $\underline{\quad}$

④ 論理演算子  
 $(a \geq 0) \text{ And } (b \geq 0)$   
 $\rightarrow a$ が0以上かつ  $b$ が0以上なら  $\underline{\quad}$

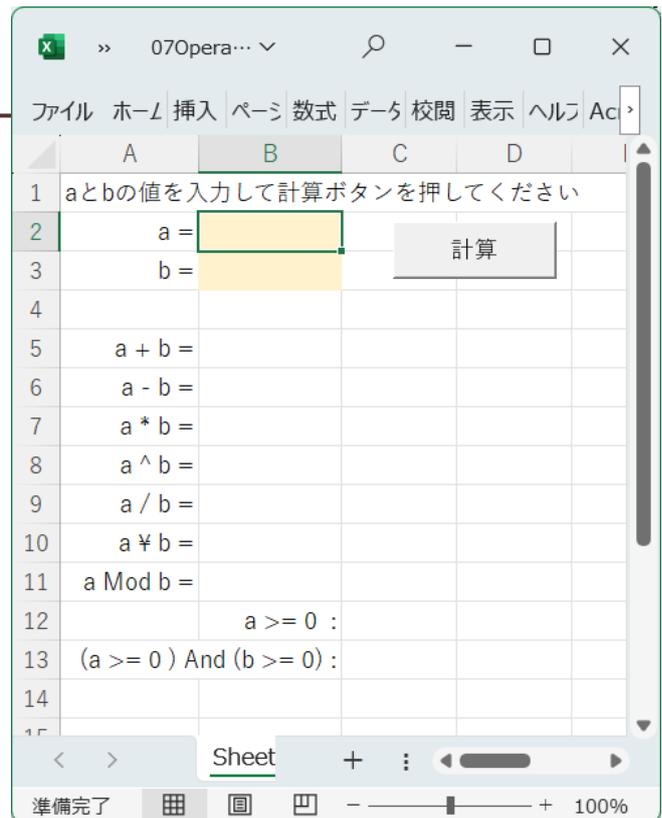
## ▶ プログラム

07Operator.xlsm

```

1 Sub ScoreArray()
2   Dim a As Long, b As Long
3   a = Cells(2, 2).Value
4   b = Cells(3, 2).Value
5
6   Cells(5, 2).Value = a + b
7   Cells(6, 2).Value = a - b
8   Cells(7, 2).Value = a * b
9   Cells(8, 2).Value = a ^ b
10  Cells(9, 2).Value = a / b
11  Cells(10, 2).Value = a \ b
12  Cells(11, 2).Value = a Mod b
13
14  Cells(12, 3).Value = a >= 0
15  Cells(13, 3).Value = (a >= 0) And (b >= 0)
16 End Sub

```





## ▶ ホワイトボード

▶ 分岐処理

① If... Then...Else ステートメント [ If文 ]

If 条件式 Then  
処理 1  
Else  
処理 2  
End If

• 処理は \_\_\_\_\_ 記述できる  
• どちらかの処理しか実行されない  
• Else ステートメントは \_\_\_\_\_ できる

② ElseIf ステートメント [ ElseIf 節 ]

If 条件式 A Then  
処理 1  
ElseIf 条件式 B Then  
処理 2  
Else  
処理 3  
End If

• ElseIf ステートメントは \_\_\_\_\_ 記述できる  
• 条件式は \_\_\_\_\_ から順に判定

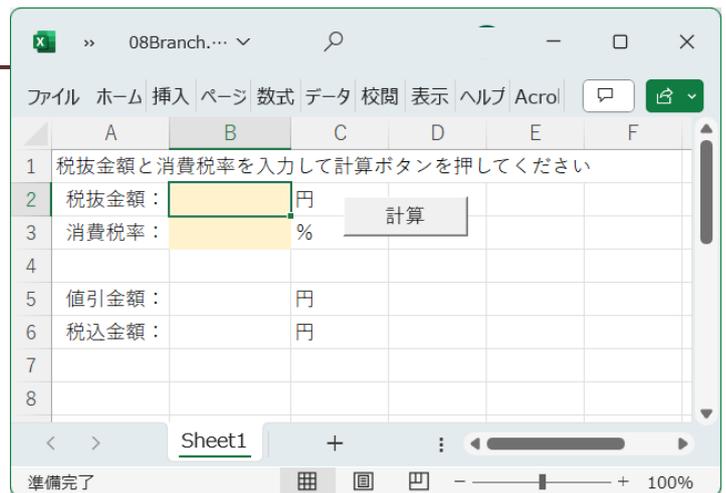
## ▶ プログラム

## 08Branch.xlsm

```

1 Sub Branch()
2   Dim price As Long      '税抜金額
3   Dim discount As Long  '値引金額
4   Dim amount As Long    '税込金額
5   Dim rate As Double    '消費税率
6
7   price = Cells(2, 2).Value
8   rate = Cells(3, 2).Value
9
10  If price >= 5000 Then
11    discount = 500
12  ElseIf price >= 3000 Then
13    discount = 300
14  Else
15    discount = 0
16  End If
17
18  amount = (price - discount) * (100 + rate) / 100
19
20  Cells(5, 2).Value = discount
21  Cells(6, 2).Value = amount
22 End Sub

```





▶ ホワイトボード

▶ 反復(繰り返し)処理

① Do...Loopステートメント [While文]  
[繰り返す \_\_\_\_\_ を指定]

Do While 条件式  
処理  
Loop

$i = 1$   
Do While  $i < 4$   
Cells(i, 1).Value = i  
 $i = i + 1$   
Loop

Flowchart for Do While: A diamond labeled '条件' (Condition). If 'True', it goes to a box labeled '処理' (Process) and loops back to the diamond. If 'False', it exits the loop.

② For...Nextステートメント [For文]  
[繰り返す \_\_\_\_\_ を指定]

For 変数名 = 開始値 To 終了値  
処理  
Next 変数名

For  $j = 1$  To 3  
Cells(j, 2).Value = j  
Next j

Flowchart for For...Next: A box labeled '変数 = 開始値' (Variable = Start Value) leads to a diamond labeled '変数 ≤ 終了値' (Variable ≤ End Value). If 'True', it goes to a box labeled '処理' (Process) and then '変数 = 変数 + 1' (Variable = Variable + 1), looping back to the diamond. If 'False', it exits the loop.

i	A	B	j
1			
2			
3			
4			

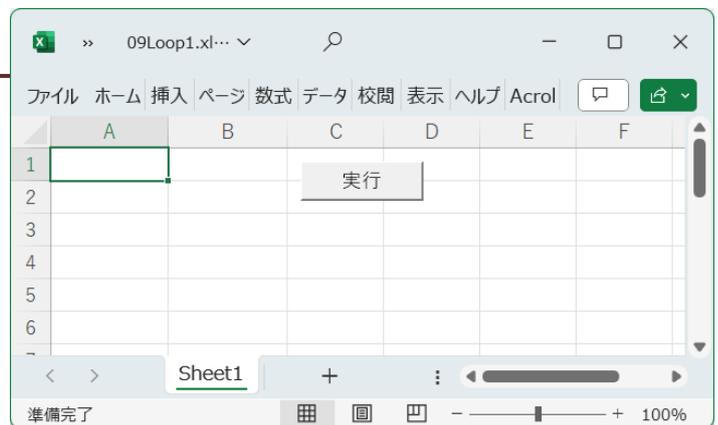
▶ プログラム

09Loop1.xlsm

```

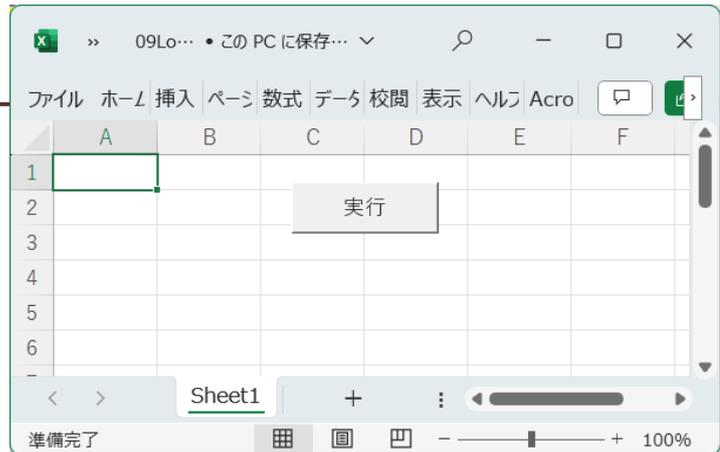
1 Sub Loop1()
2   Dim i As Long, j As Long
3   i = 1
4   Do While i < 4
5     Cells(i, 1).Value = "While : " & i
6     i = i + 1
7   Loop
8
9   For j = 1 To 3
10    Cells(j, 2).Value = "For : " & j
11  Next j
12 End Sub

```



## 09Loop2.xlsm

```
1 Sub ScoreArray()  
2   Dim score(2) As Long  
3   score(0) = 80  
4   score(1) = 100  
5   score(2) = 90  
6  
7   Dim name As Variant  
8   name = Array("菅原", "桜井", "田原")  
9  
10  Dim sum_score As Long  
11  sum_score = 0  
12  For i = 0 To 2  
13    Cells((i + 1), 1).Value = name(i)  
14    Cells((i + 1), 2).Value = score(i)  
15    sum_score = sum_score + score(i)  
16  Next i  
17  
18  Dim avg As Double  
19  avg = sum_score / 3  
20  
21  Cells(5, 1).Value = "合計"  
22  Cells(5, 2).Value = sum_score  
23  Cells(6, 1).Value = "平均"  
24  Cells(6, 2).Value = avg  
25 End Sub
```





## ▶ ホワイトボード

▶ 関数

- ① \_\_\_\_\_ : 特定の処理をまとめたもの
- ② Sub プロシージャ < \_\_\_\_\_ あり >  
Sub プロシージャ名 (引数, 引数, ...)  
    処理  
End Sub
- ③ Function プロシージャ < \_\_\_\_\_ あり >  
Function プロシージャ名 (引数, 引数, ...)  
    処理  
End Function

- \_\_\_\_\_ : 呼び出し元から受け取る値
- \_\_\_\_\_ : 呼び出し元に返す値  
返す値は \_\_\_\_\_ に代入

```

Call ShowMsg

Call Calc(90, 80)

status =
  Judge(90, 80)
  
```

```

Sub ShowMsg()
  MsgBox("集計します")
End Sub

Sub Calc(e, m)
  Cells(-).Value = (e+m) / 2
End Sub

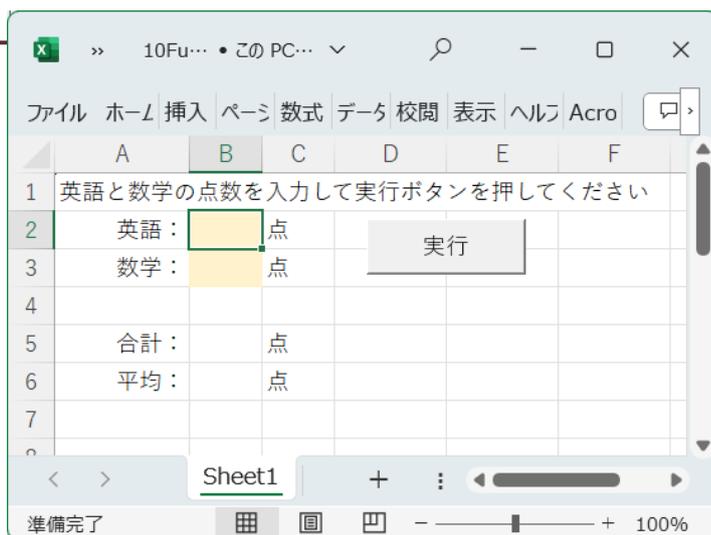
Function Judge(e, m)
  If (e >= 80) Then
    Judge = "合格!"
  Else
    Judge = "不合格"
  End If
End Function
  
```

## ▶ プログラム

### 10Function.xlsm

```

1 Sub MyFunction()
2   Dim eng As Long, math As Long
3   Dim status As String
4   Call ShowMsg
5   eng = Cells(2, 2).Value
6   math = Cells(3, 2).Value
7   Call Calc(eng, math)
8   status = Judge(eng, math)
9   Cells(6, 4).Value = status
10 End Sub
11 Sub ShowMsg()
12   MsgBox ("テスト結果を集計します")
13 End Sub
14 Sub Calc(e, m)
15   Cells(5, 2).Value = e + m
16   Cells(6, 2).Value = (e + m) / 2
17 End Sub
18 Function Judge(e, m)
19   If ((e >= 80) And (m >= 80)) Then
20     Judge = "合格!"
21   Else
22     Judge = "不合格"
23   End If
24 End Function
  
```





## ▶ ホワイトボード

▶ ライブラリ

① \_\_\_\_\_ : よく使う処理を利用しやすくまとめたもの  
(開発画面で) [ツール]-[参照設定]で指定

代表的なライブラリ

- ④ Visual Basic for Applications  
VBAの基本機能 (MsgBox関数, Array関数など)
- ⑤ Microsoft Excel Object Library  
Excel関連の機能 (Range, Cellsなど)
- ⑥ Microsoft Office Object Library  
Officeアプリ関連の機能

② \_\_\_\_\_ : VBAで提供されている関数

- MsgBox("文字列") :  
ダイアログボックスに文字列を表示
- Array(値, 値, ...):  
引数を要素にもつ配列を取得
- Rnd(): 0以上1未満の乱数を取得
- Int(数値): 数値の整数部分を取得

③ \_\_\_\_\_ (Application Programming Interface)  
プログラムやサービスの機能を外部から使う仕組みや利用方法

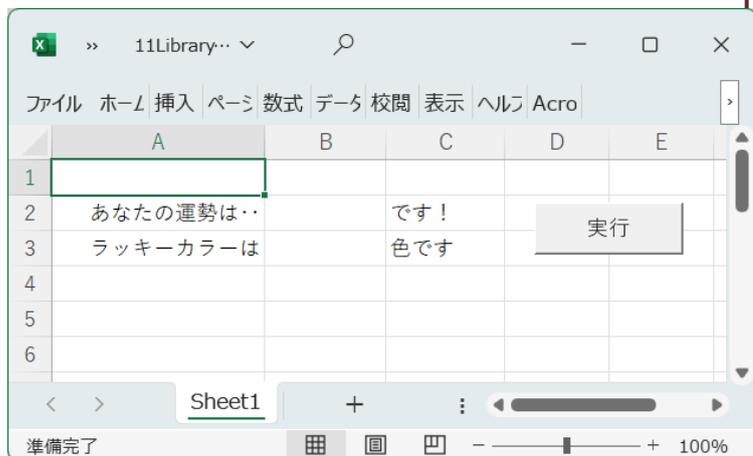
## ▶ プログラム

### 11Library.xlsm

```

1 Sub UseLibrary()
2   Dim fn1 As Double
3   Dim fn2 As Long
4   Dim judge As String '運勢
5   Dim color As String 'ラッキーカラー
6
7   fn1 = Rnd()
8   If fn1 >= 0.7 Then
9     judge = "大吉"
10  Elseif fn1 >= 0.4 Then
11    judge = "中吉"
12  Elseif fn1 >= 0.1 Then
13    judge = "小吉"
14  Else
15    judge = "凶"
16  End If
17
18  Dim colors As Variant
19  colors = Array("赤", "黄", "黄緑", "緑", "水", _
20               "青", "紫", "白", "黒", "金")
21  fn2 = Int(Rnd() * 10)
22  color = colors(fn2)
23
24  Cells(2, 2).Value = judge
25  Cells(3, 2).Value = color
26  ' Cells(4, 2).Value = fn1
27  ' Cells(5, 2).Value = fn2
28 End Sub

```





## ▶ ホワイトボード

▶ モデル化とシミュレーション

問題発生  $\rightarrow$  \_\_\_\_\_ : 対象を単純化して  
模型や数式、図で表現  $\rightarrow$  \_\_\_\_\_ :  
モデルを使って試行  $\rightarrow$  解決

手順 ① \_\_\_\_\_ の明確化  
② 構成要素の \_\_\_\_\_  
③ 数式や図で \_\_\_\_\_  
④ \_\_\_\_\_  
⑤ モデルを \_\_\_\_\_  
⑥ 問題を \_\_\_\_\_

表現方法?  $\rightarrow$  物理モデル / 論理モデル

特性?  $\rightarrow$  実物モデル / 拡大モデル / 縮小モデル / 動的モデル / 静的モデル / 確率モデル / 確定モデル / 数式モデル / 図的モデル / 連続モデル / 離散モデル

$F=ma$   
時間変化? / 不確定要素? / 連続データ?

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 対象を単純化して模型や数式、図などで表現することを(① \_\_\_\_\_)といい、モデルを用いて実際の現象や実物の動作を試すことを(② \_\_\_\_\_)という。
- ② (①)するときには、まず発生している(③ \_\_\_\_\_)を明確化する。次にモデルの構成要素を(④ \_\_\_\_\_)し、数式や図で表現する。その後、モデルを(⑤ \_\_\_\_\_)しながら(②)を繰り返すことによって、より良い解決方法を発見できる。

①モデル化 ②シミュレーション ③目的 ④分析 ⑤修正  
(↑紙面の都合でこのページに記載)



## ▶ ホワイトボード

▶ シミュレーション①：つり銭問題

STEP 1 問題の明確化  
文化祭で200円の商品 → 100 何枚必要？

STEP 2 構成要素の分析  
100 の \_\_\_\_\_、500 の \_\_\_\_\_  
支払い(200円/500円)の \_\_\_\_\_、\_\_\_\_\_

STEP 3 数式や図で表現  
・ 200円/500円は \_\_\_\_\_ % ・ 来客は \_\_\_\_\_ 名  
・ { 乱数が 0.0 ~ 0.5 → 200円 → 100 \_\_\_\_\_ 枚  
0.5 ~ 1.0 → 500円 → 100 \_\_\_\_\_ 枚

STEP 4 シミュレーション

```

coin = 0
For i = 1 To 50
  rn = Rnd()
  If (rn < 0.5) Then
    coin = coin + 2
  Else
    coin = coin - 3
  End If
  Cells(-).Value = coin & "枚"
Next i

```

5 モデルを修正 6 問題を解決  
・ 支払いの \_\_\_\_\_ ・ \_\_\_\_\_ を用意  
・ \_\_\_\_\_ は? ・ 紙幣は \_\_\_\_\_

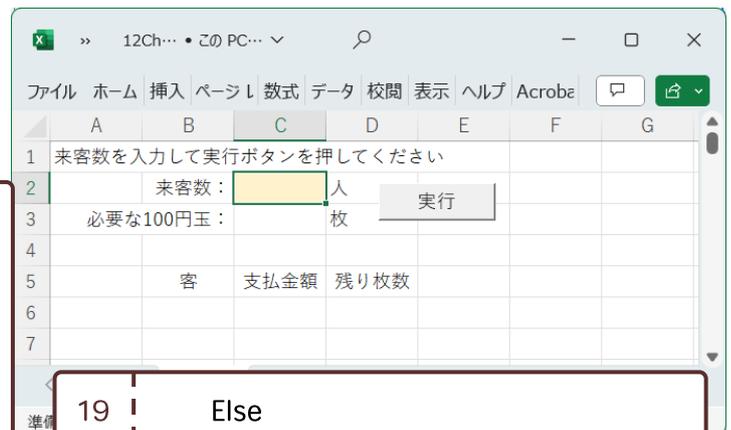
## ▶ プログラム

## 12Change.xlsm

```

1 Sub Change()
2   Dim coin As Long, min_coin As Long
3   Dim pay As String
4   Dim num As Long
5
6   coin = 0      '100 円玉の枚数
7   min_coin = 0 '最大不足枚数
8   pay = ""     '支払金額
9   num = Cells(2, 3).Value '来客数
10
11  Dim i As Long
12  Dim rn As Double
13
14  For i = 1 To num
15    rn = Rnd()
16    If (rn < 0.5) Then
17      pay = "200 円"
18      coin = coin + 2

```



```

19 Else
20   pay = "500 円"
21   coin = coin - 3
22 End If
23 Cells(i + 5, 2).Value = i
24 Cells(i + 5, 3).Value = pay
25 Cells(i + 5, 4).Value = coin & "枚"
26 If (coin < min_coin) Then
27   min_coin = coin
28 End If
29 Next i
30
31 Cells(3, 3).Value = Abs(min_coin)
32 End Sub

```



## ▶ ホワイトボード

▶ シミュレーション②：待ち行列

STEP 1 問題の明確化  
バイトで待ち行列 → 待ち時間？ 短縮時間？

STEP 2 構成要素の分析  
\_\_\_\_\_ ( \_\_\_ 台), \_\_\_\_\_ ( \_\_\_ 秒)  
\_\_\_\_\_ ( \_\_\_ 人), \_\_\_\_\_ ( \_\_\_ 秒)

STEP 3 数式や図で表現  
☐ 待ちなし      ☑ 待ちあり

A B

C D

• start = \_\_\_\_\_      • start = \_\_\_\_\_

STEP 4 シミュレーション

```

start = 0      arrival = 0
finish = 0     wait = 0
For i = 1 To 100
  rn = Int(Rnd() * 100)
  arrival = arrival + rn
  If (finish <= arrival) Then
    start = arrival
  Else
    start = finish
  End If
  wait = start - arrival
  finish = start + 30
  (各変数の値をセルに出力)
Next i

```

5 モデルを修正      6 問題を解決  
• サービス時間を \_\_\_\_\_      • レジを \_\_\_\_\_

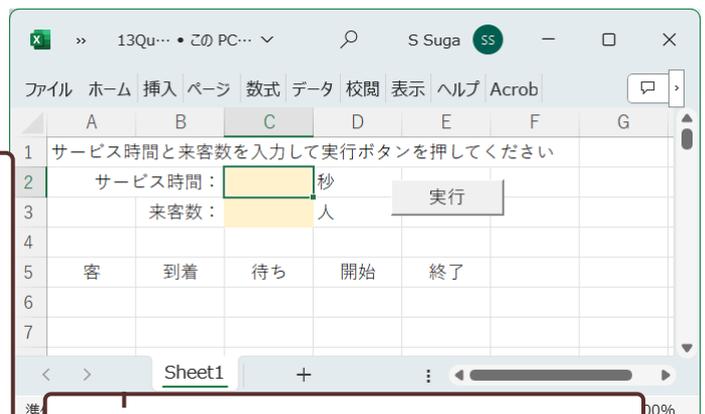
## ▶ プログラム

## 13Queue.xlsm

```

1 Sub Queue()
2   Dim start As Long, finish As Long
3   Dim arrival As Long, wait As Long
4   start = 0   '開始時刻
5   finish = 0  '終了時刻
6   arrival = 0 '到着時刻
7   wait = 0   '待ち時間
8
9   Dim service As Long, num As Long
10  service = Cells(2, 3).Value 'サービス時間
11  num = Cells(3, 3).Value     '来客数
12
13  Dim i As Long, rn As Long
14
15  For i = 1 To num
16    rn = Int(Rnd() * 100)
17    arrival = arrival + rn
18    If (finish <= arrival) Then

```



```

19     start = arrival
20   Else
21     start = finish
22   End If
23   wait = start - arrival
24   finish = start + service
25   Cells(i + 5, 1).Value = i
26   Cells(i + 5, 2).Value = arrival
27   Cells(i + 5, 3).Value = wait
28   Cells(i + 5, 4).Value = start
29   Cells(i + 5, 5).Value = finish
30 Next i
31 End Sub

```

# 情報通信ネットワークとデータの活用



【4章】ネットワークやデータベースを基本から解説

 <https://youtu.be/o9YLtvlk8oc>

動画時間：3:39:35

学習項目：21



## 4章のゴール

- 情報通信ネットワークを構成する機器やデータが転送される仕組みを理解しよう
- 情報通信ネットワークを用いた情報システムの仕組みや情報セキュリティを確保するために用いられている技術を理解しよう
- 情報システムで用いられているデータベース技術を理解しよう
- データを収集・整理・分析する手法を知り、情報社会の問題を発見したり解決したりできる力を身につけよう

## ▶ キーワード

- |  |  |                                      |
|--|--|--------------------------------------|
| <input type="checkbox"/> 情報通信ネットワーク      | <input type="checkbox"/> パリティチェック      | <input type="checkbox"/> 暗号化・復号      |
| <input type="checkbox"/> LAN・WAN         | <input type="checkbox"/> WWW・Web       | <input type="checkbox"/> 共通鍵暗号方式     |
| <input type="checkbox"/> インターネット         | <input type="checkbox"/> Webサーバ        | <input type="checkbox"/> 公開鍵暗号方式     |
| <input type="checkbox"/> プロトコル           | <input type="checkbox"/> Webブラウザ       | <input type="checkbox"/> 秘密鍵・公開鍵     |
| <input type="checkbox"/> 有線・無線LAN        | <input type="checkbox"/> HTML          | <input type="checkbox"/> デジタル署名      |
| <input type="checkbox"/> ハブ・スイッチ・ルータ     | <input type="checkbox"/> クライアントサーバシステム | <input type="checkbox"/> 電子認証        |
| <input type="checkbox"/> Wi-Fi・Bluetooth | <input type="checkbox"/> URL           | <input type="checkbox"/> 情報システム      |
| <input type="checkbox"/> TCP/IP 4階層モデル   | <input type="checkbox"/> ドメイン名         | <input type="checkbox"/> データベース・DBMS |
| <input type="checkbox"/> HTTP            | <input type="checkbox"/> DNS           | <input type="checkbox"/> データモデル      |
| <input type="checkbox"/> TCP・UDP         | <input type="checkbox"/> 情報セキュリティ      | <input type="checkbox"/> リレーショナルモデル  |
| <input type="checkbox"/> IP・IPアドレス       | <input type="checkbox"/> 機密性・完全性・可用性   | <input type="checkbox"/> 尺度水準        |
| <input type="checkbox"/> ルーティング          | <input type="checkbox"/> 認証            | <input type="checkbox"/> 回帰分析        |
| <input type="checkbox"/> ルーティングテーブル      | <input type="checkbox"/> ファイアウォール      | <input type="checkbox"/> テキストマイニング   |



## ▶ ホワイトボード

▶ ネットワークの基本

- \_\_\_\_\_ ( \_\_\_\_\_ ) : コンピュータなどの機器をつないで情報をやり取りする通信網
- ① \_\_\_\_\_ : 限られた範囲内のNW
- ② \_\_\_\_\_ : 遠隔地をつなぐNW
  - \_\_\_\_\_ : 回線を占有
  - \_\_\_\_\_ : 回線を共有
- ③ \_\_\_\_\_ : 世界中のコンピュータやNWをつないだNW. \_\_\_\_\_ と契約必要
- NWの構成要素 : NW機器 ( \_\_\_\_\_ )、アプリ ( \_\_\_\_\_ )、通信ルール ( \_\_\_\_\_ )

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 家や学校などの限られた範囲のネットワークを( ① \_\_\_\_\_ )といい、(①)同士を広い範囲でつないだネットワークを( ② \_\_\_\_\_ )という。(①)や(②)がつながり世界規模に発展したネットワークが( ③ \_\_\_\_\_ )である。
- ② ネットワークは機器類の( ④ \_\_\_\_\_ )とアプリケーションなどの( ⑤ \_\_\_\_\_ )、通信ルールをまとめたプロトコルで構成される。



▶ ホワイトボード

▶ 有線 LAN (イーサネット)

- \_\_\_\_\_ : 限られた範囲内でケーブルをつないで通信するネットワーク → 無線より \_\_\_\_\_ & \_\_\_\_\_  
主流の規格(ケーブルや通信方法の標準ルール)は \_\_\_\_\_

<p>[ケーブル]</p> <p>① _____ ケーブル : 銅線を絶縁体 _____ などで覆う → △ 太くて不便 </p> <p>② _____ ケーブル (= _____ ケーブル) : 8本の銅線を2本ずつより合わせる → ○ ノイズに強い、細くて柔軟、安価</p> <p>③ _____ ケーブル : ガラスやプラスチックで 光信号を送る → ○ 高速 △ 破損、高価</p>	<p>[機器]</p> <p>① _____ : LANを他のネットワークに接続</p> <p>② 集線装置 : 機器同士を接続</p> <ul style="list-style-type: none"> <li>— _____ (リピータハブ) : 全ポートに送信</li> <li>— _____ (スイッチングハブ) : 宛先に送信</li> </ul> 
--	--

▶ 確認テスト

画面を止めて考えてみよう!

① 限られた範囲でケーブルをつないで通信するネットワークを  
(① \_\_\_\_\_)という。今は(② \_\_\_\_\_)と  
いう規格が主流であり、8本の銅線を2本ずつより合わせてま  
とめた(③ \_\_\_\_\_)ケーブルがよく用いられる。

② (①)で使われる集線装置には、届いた信号を全ポートに送  
信する(④ \_\_\_\_\_)や宛先のポートのみに送信をする(⑤  
\_\_\_\_\_ )がある。

- ①LAN ②WAN ③インターネット ④ハードウェア ⑤ソフトウェア



▶ ホワイトボード

▶ 無線 LAN

- \_\_\_\_\_ : 限られた範囲内で電波を使って通信するネットワーク  
→ データを盗み見されないように \_\_\_\_\_ でセキュリティを確保!

[無線通信の技術]

	Wi-Fi	Bluetooth	[無線LANの規格]			
	IEEE 802.11	IEEE 802.15.1	名前	略称	周波数 [GHz]	速度 [bps]
規格	( ~ m, ~ bps )	( ~ m, ~ bps )	IEEE 802.11a	x	5	54M
			: b	x	2.4	11M
			: g	x	2.4	54M
			: n	Wi-Fi 4	2.4/5	600M
			: ac	: 5	5	6.9G
			: ax	: 6	2.4/5	9.6G
用途						

▶ 確認テスト

画面を止めて考えてみよう!

- 限られた範囲でケーブルを直接つながないで通信するネットワークを(① \_\_\_\_\_)という。便利な反面、セキュリティを確保するために(② \_\_\_\_\_)や(③ \_\_\_\_\_)などの対策をとる必要がある。
- 無線通信の手段として、遠距離に高速で通信できるIEEE 802.11規格に基づく(④ \_\_\_\_\_)や、省電力で接続がしやすい(⑤ \_\_\_\_\_)がよく使われている。

①有線 LAN ②イーサネット ③ツイストペア ④ハブ ⑤スイッチ



## ▶ ホワイトボード

▶ TCP/IP 4階層モデル

- \_\_\_\_ (通信規約) : 通信の手順やデータ形式などに  
関する約束ごと。 \_\_\_\_ してまとめられている
- \_\_\_\_ 4階層モデル : インターネットの通信をモデル化

表現	友達	家族	先生
言語	日本語	英語	
媒体	電話	手紙	メール

4層	____ 層	アプリケーション間の通信ルール → アプリケーション同士で ____	HTTP, POP SMTP, FTP	ヘッダ データ 成功/失敗、文字コードなど
3層	____ 層	エラー検出・訂正などの通信制御 → データが ____ に届く	UDP	ヘッダ データ データの順番、正しいかの判別情報
2層	____ 層	データの転送経路を選択 → ____ 宛先にデータが届く		ヘッダ データ 宛先・送信元のアドレス情報
1層	____ 層	物理的に接続 → ____ する機器にデータが届く	イーサネット	ヘッダ データ NW機器や信号などの情報

## ▶ 確認テスト

画面を止めて考えてみよう!

通信における約束ごとを(① \_\_\_\_\_) (通信規約) という。インターネットで用いられる(①)は、以下のような(② \_\_\_\_\_) 4階層モデルで表現される。

- 4層:③ \_\_\_\_\_ 層:アプリケーション間の通信
- 3層:④ \_\_\_\_\_ 層:データを正確に届ける
- 2層:⑤ \_\_\_\_\_ 層:データを遠い宛先に届ける
- 1層:⑥ \_\_\_\_\_ 層:データを隣接機器に届ける

①無線 LAN ②認証 ③暗号化 ④Wi-Fi ⑤Bluetooth



## ▶ ホワイトボード

▶ IPアドレス

- \_\_\_\_\_ : TCP/IPネットワークで各コンピュータを識別するための固有のアドレス番号

IPv4: ビット → 2 (約 \_\_\_\_\_) 通り

- 8ビットずつ \_\_\_\_\_ で区切って \_\_\_\_\_ 進数表記
- ネットワークを識別する \_\_\_\_\_ 部と  
各機器を識別する \_\_\_\_\_ 部で構成
- ホスト部 すべて 0 → \_\_\_\_\_ を表現  
: すべて 1 → \_\_\_\_\_ の機器を表現

[ \_\_\_\_\_ IPアドレス : 世界で唯一  
: \_\_\_\_\_ : LAN内で使用

IPv6: ビット → 2 (約 340 滴) 通り

1100 0000	1010 1000	0000 0000	0000 1001
.			

クラスA	0	7ビット	24ビット (約1,600万)
B	10	14ビット	16ビット (65,534)
C	110	21ビット	8ビット (254)

## ▶ 確認テスト

画面を止めて考えてみよう!

- インターネットではデータの宛先を指定するために(① \_\_\_\_\_)が使われる。IPv4では(①)は(② \_\_\_\_\_)ビットで構成され、(③ \_\_\_\_\_)ビットずつ「.」で区切って10進数で表記されることが多い。
- インターネット上で重複なく割り当てられているIPアドレスを(④ \_\_\_\_\_)IPアドレスといい、LANなどで自由に使えるアドレスを(⑤ \_\_\_\_\_)IPアドレスという。

- ①プロトコル    ②TCP/IP    ③アプリケーション    ④トランスポート  
⑤インターネット    ⑥ネットワークインタフェース



## ▶ ホワイトボード

▶ ルーティング

- \_\_\_\_\_ (経路制御): IPの役割の1つ、宛先までの最適な経路を判断してデータを届ける経路は各機器の \_\_\_\_\_ (経路制御表) をもとに決められる

[ルータ①のルーティングテーブル]

宛先	ゲートウェイ	インタフェース	メトリック
192.168.1.0	直接	192.168.1.1	
192.168.3.0	直接	192.168.3.1	

[ルータ②のルーティングテーブル]

宛先	ゲートウェイ	インタフェース	メトリック
192.168.2.0	直接	192.168.2.1	
192.168.3.0	直接	192.168.3.1	

## ▶ 確認テスト

画面を止めて考えてみよう!

データを目的のコンピュータまで、最適な経路を判断・中継して届けることを(① \_\_\_\_\_) (経路制御)という。(①)の経路は各コンピュータやルータが持つ(② \_\_\_\_\_) (経路制御表)によって決められる。

(②)にはデータを次に届ける機器を表す(③ \_\_\_\_\_)や、データの出力先を表す(④ \_\_\_\_\_)などの情報がまとめられている。

- ①IP アドレス ②32 ③8 ④グローバル ⑤プライベート



## ▶ ホワイトボード

▶ 通信の信頼性

★ 通信時はデータのビット \_\_\_\_\_ や \_\_\_\_\_ が発生する可能性有  
 → 受信側で誤りを \_\_\_\_\_ して、 \_\_\_\_\_ や \_\_\_\_\_ 依頼の仕組みが大切!

① (奇偶検査)  
 「1」の個数を調べて誤りを \_\_\_\_\_ (奇数パリティ)

データ: 1001100

奇数パリティ → \_\_\_\_\_ を付加  
 偶数 → \_\_\_\_\_

受信後: 1011100 → 誤り  
 1011101 → 誤り

② TCPの通信制御  
 データの \_\_\_\_\_ を管理して  
 不達時は \_\_\_\_\_ を依頼

送信 (3000バイト) → 受信 (1 (1000))  
 1 OK!  
 2 OK!  
 3 OK!

■ \_\_\_\_\_ : データの誤り検出のために元データに付加される数字、バーコードやマイナンバーなどで活用.

## ▶ 確認テスト

画面を止めて考えてみよう!

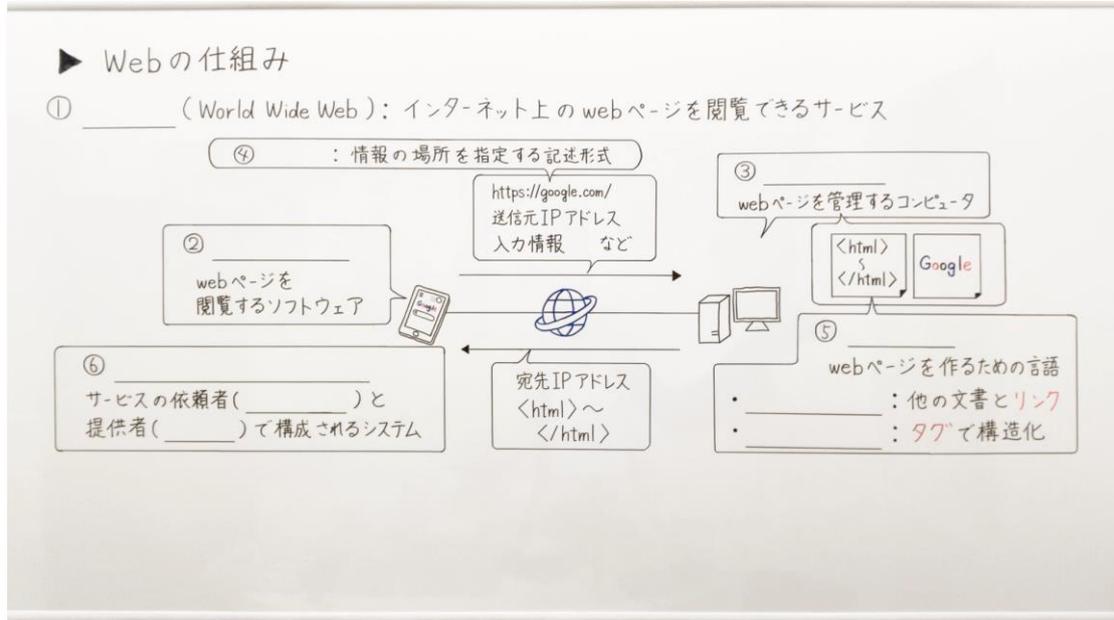
① ビットの誤り検査で使われる「一定のビット列の中に1が奇数(偶数)個あること」を示すビットを(① \_\_\_\_\_)といい、(①)を用いた誤り検出を(② \_\_\_\_\_)という。8ビットの奇数パリティでデータ1001100を送信する場合、(①)の値は(③ \_\_\_\_\_)である。

② TCPプロトコルは受信時にデータの(④ \_\_\_\_\_)を管理して、データが届かないときは(⑤ \_\_\_\_\_)を依頼する。

- ①ルーティング ②ルーティングテーブル ③ゲートウェイ ④インタフェース



## ▶ ホワイトボード



## ▶ 確認テスト

画面を止めて考えてみよう!

- ① (① \_\_\_\_\_)はインターネット上のwebページを閲覧できるサービスである。webページのデータは(② \_\_\_\_\_)上で管理され、ユーザが閲覧するときは(③ \_\_\_\_\_)というソフトウェアを使ってwebページの場所を(④ \_\_\_\_\_)で指定する。
- ② webページは他の文書へのリンクを埋め込むことができ、タグで構造化する(⑤ \_\_\_\_\_)という言葉で記述する。

①パリティビット ②パリティチェック ③0 ④順序 ⑤再送



## ▶ ホワイトボード

▶ URL と DNS の仕組み

① \_\_\_\_\_ : 情報の場所を指定

② \_\_\_\_\_ : インターネット上のコンピュータやネットワークを識別する名前 → \_\_\_\_\_ や \_\_\_\_\_ で使用  
★ 実は \_\_\_\_\_ を変換したもの

③ \_\_\_\_\_ : ドメイン名と IP アドレスを対応づけて変換 ( \_\_\_\_\_ ) する仕組み

◀ プロトコル ▶

http → web ページ

https → : (暗号化)

ftp → ファイル

◀ トップレベルドメイン ▶

ccTLD	gTLD
jp 日本	com 商業組織
ca カナダ	edu 教育機関

https:// www.google.com/ index.html

スキーム名    ホスト名    ドメイン名    パス名

① www.google.com の IP?

② 172.217.175.4

③ https://www.google.com/index.html

④ <http> ~</http>

## ▶ 確認テスト

画面を止めて考えてみよう!

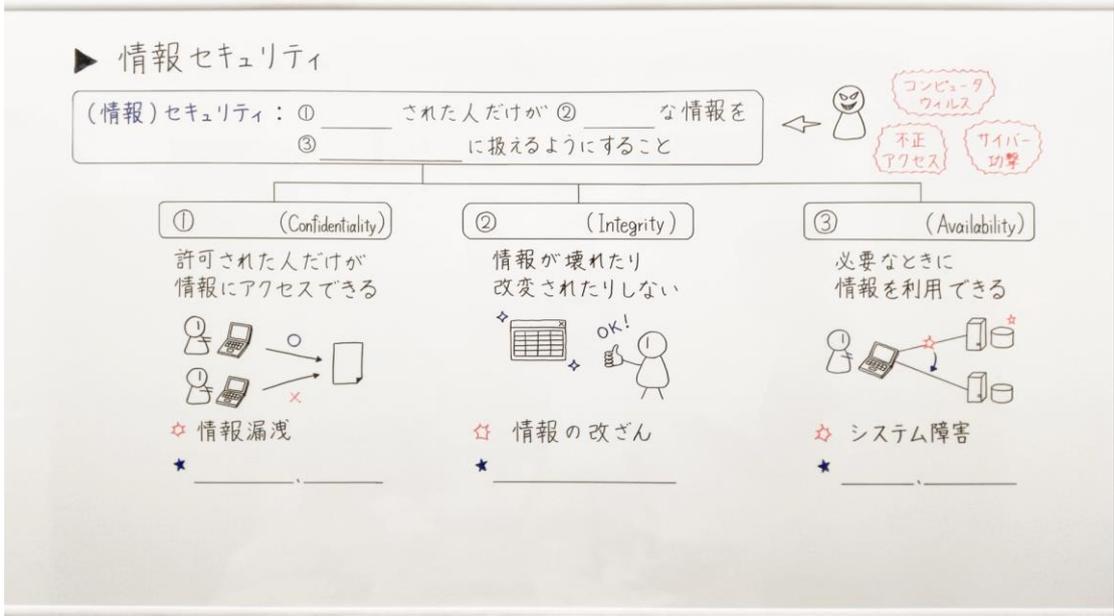
web ページを閲覧するときは、見たいページの場所を ( ① \_\_\_\_\_ ) で指定する。(①)にはプロトコル名やwebサーバのIPアドレスをわかりやすい名前に置き換えた ( ② \_\_\_\_\_ ) などが含まれる。

(②)の一番右の部分は ( ③ \_\_\_\_\_ ) と呼ばれる。(②)とIPアドレスを相互に変換する仕組みのことを ( ④ \_\_\_\_\_ ) という。

- ①WWW (web など)    ②web サーバ    ③web ブラウザ    ④URL    ⑤HTML



## ▶ ホワイトボード



## ▶ 確認テスト

画面を止めて考えてみよう!

情報セキュリティは次の3つの性質を保つことを指す。

- (① \_\_\_\_\_) : 許可された人だけがアクセスできる
- (② \_\_\_\_\_) : 情報が壊れたり変更されたりしない
- (③ \_\_\_\_\_) : 必要などきに利用できる

(①)が不足していると(④ \_\_\_\_\_)が発生したり、(②)が不足していると情報が(⑤ \_\_\_\_\_)されたりしてしまうリスクがある。

①URL ②ドメイン名 ③トップレベルドメイン ④DNS



## ▶ ホワイトボード

▶ 認証とアクセス制御

情報の \_\_\_\_\_ (許可された人だけアクセス可) を高める技術 → \_\_\_\_\_ など

① ユーザ認証  
ユーザが正規の利用者かを確認

- ユーザID & パスワード
- \_\_\_\_\_ : 個人識別番号
- \_\_\_\_\_ 認証 : 携帯宛てコード
- ICカード 認証
- \_\_\_\_\_ 認証

× 個人情報  
× 英字(数字)のみ  
× 使いまわし

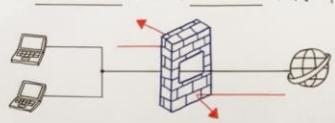
↓

◎ 英・数・記号 10桁↑  
◎ 土台ワードを決定  
→ 置換と追加

(盗み見など) に注意!

② アクセス制御  
特定の利用者だけが  
データを扱えるように制限

- \_\_\_\_\_ : ネットワークの  
出入口に設置する装置 or ソフトウェア
- \_\_\_\_\_ をみて \_\_\_\_\_ の判断



## ▶ 確認テスト

画面を止めて考えてみよう!

- ① サービスを利用するときに、ユーザが正規の利用者かを確認することを(① \_\_\_\_\_)という。ユーザIDとパスワードが使われることが多いが、個人識別番号の(② \_\_\_\_\_)や指紋や顔を用いる(③ \_\_\_\_\_)などの方法もある。
- ② ネットワークの外部からのアクセスを制御するために、ネットワークの出入口に設置する装置やソフトウェアのことを(④ \_\_\_\_\_)という。

①機密性 ②完全性 ③可用性 ④情報漏洩 ⑤改ざん



## ▶ ホワイトボード

▶ 暗号化の仕組み

\_\_\_\_\_ : 情報を送るときに、盗聴されても解読できない形式に変換すること

① 共通鍵暗号方式  
送・受 同じ鍵( \_\_\_\_\_ )を使用

② 公開鍵暗号方式  
送 \_\_\_\_\_ で暗号化 受 \_\_\_\_\_ で復号

○ 処理が速い  
△ \_\_\_\_\_ の共有方法、相手ごとに \_\_\_\_\_ が必要

○ 安全性が高い、鍵を管理しやすい  
△ 処理が遅い

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 情報を送るときに第三者に盗聴されても解読できないようにすることを( ① \_\_\_\_\_ )という。(①)する前のデータを( ② \_\_\_\_\_ )といい、(①)されたデータを( ③ \_\_\_\_\_ )という。(③)を(②)に戻すことを( ④ \_\_\_\_\_ )という。
- ② 暗号化の方式には(①)と(④)に同じ鍵を使う( ⑤ \_\_\_\_\_ )暗号方式と、送信者が公開鍵を使って(①)して受信者が秘密鍵を使って(④)する( ⑥ \_\_\_\_\_ )暗号方式がある。

①認証 ②PIN ③生体認証 ④ファイアウォール



## ▶ ホワイトボード

▶ デジタル署名と電子認証

- \_\_\_\_\_ : 公開鍵暗号の技術を使ってデータの改ざんがないことを証明

**送**

- ① \_\_\_\_\_ で文書から \_\_\_\_\_ を取得
- ② ①を \_\_\_\_\_ で暗号化 → \_\_\_\_\_
- ③ 文書、②、\_\_\_\_\_ を送付

**受**

- ④ (①と同じ)
- ⑤ デジタル署名を \_\_\_\_\_ で復号 ← ⑥

\_\_\_\_\_ : 認証局が発行する電子証明書でデジタル署名が本人のものであることを証明

\_\_\_\_\_ : 暗号化や電子認証を用いる通信技術. URL が https://~となる.

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 公開鍵暗号方式の技術を用いてデータの改ざんがないことを証明する方法に(① \_\_\_\_\_)がある。送信者は文書から(② \_\_\_\_\_)で要約文を作り、さらに(③ \_\_\_\_\_)で暗号化した(①)を文書と一緒に送信する。
- ② (①)が送信者本人のものであることを証明する電子データを(④ \_\_\_\_\_)といい、(④)により発信者が本人であることを証明する仕組みを(⑤ \_\_\_\_\_)という。

①暗号化 ②平文 ③暗号文 ④復号 ⑤共通鍵 ⑥公開鍵



## ▶ ホワイトボード

▶ 情報システム

\_\_\_\_\_ : 情報機器をネットワークでつないで全体で情報処理をする仕組み

[身近な例]

- \_\_\_\_\_ : 入金や出金、振込などができる端末機、銀行やコンビニなどに設置
- \_\_\_\_\_ : 人工衛星を使って地球上の位置を特定するシステム
- \_\_\_\_\_ : 高速道路の料金支払を自動化するシステム
- \_\_\_\_\_ : 商品の販売数や購売情報を管理するシステム

[購売情報の収集と活用]

→ 日々蓄積される膨大なデータ( \_\_\_\_\_ )をどのように活用するかが大切!

## ▶ 確認テスト

画面を止めて考えてみよう!

情報機器をネットワークでつないで、全体としてまとまって動く仕組みを(① \_\_\_\_\_)という。銀行の(② \_\_\_\_\_)や、人工衛星を用いて位置を特定する(③ \_\_\_\_\_)、高速道路の料金支払を行う(④ \_\_\_\_\_)などがある。

コンビニでは商品の販売数や購買情報を(⑤ \_\_\_\_\_)で管理している。(⑤)で蓄積される(⑥ \_\_\_\_\_)は、その後の販売戦略や商品企画に活用されている。

- ①デジタル署名 ②ハッシュ関数 ③秘密鍵 ④電子証明書 ⑤電子認証



## ▶ ホワイトボード

▶ データベースの基本

- \_\_\_\_\_ ( \_\_\_\_\_ ): ある目的のためにデータを整理・蓄積したもの
- \_\_\_\_\_ ( \_\_\_\_\_ ): データベースを管理するソフトウェア

- ① \_\_\_\_\_ : データの重複や、不正なデータの登録や更新を防ぐ
- ② \_\_\_\_\_ : 複数のユーザが同時に操作しても矛盾が生じないようにする
- ③ \_\_\_\_\_ : ユーザ認証をしたりアクセス権を設定したりできる
- ④ \_\_\_\_\_ : 障害に備えてバックアップしたり障害時にリストアしたりできる

## ▶ 確認テスト

画面を止めて考えてみよう!

ある目的のためにデータを整理・蓄積したものを(① \_\_\_\_\_ )という。(①)を効率良く使うために管理するソフトウェアを(② \_\_\_\_\_ )といい、(②)には次のような機能がある。

- (③ \_\_\_\_\_ ): データの重複や不正なデータを防ぐ
- (④ \_\_\_\_\_ ): 複数のユーザでデータを操作できる
- (⑤ \_\_\_\_\_ ): 認証したりアクセス権を設定したりできる
- (⑥ \_\_\_\_\_ ): 障害時に備えてバックアップなどをする

①情報システム ②ATM ③GPS ④ETC ⑤POS ⑥ビッグデータ



## ▶ ホワイトボード

▶ データモデル

- \_\_\_\_\_ : データベース内のデータの表現形式

モデル

```

graph TD
    商品 --> スマホ
    商品 --> PC
    商品 --> マウス
    スマホ --> A社
    スマホ --> B社
    PC --> A社
    PC --> C社
    マウス --> B社
    マウス --> D社
          
```

モデル

```

graph TD
    商品 --> スマホ
    商品 --> PC
    商品 --> マウス
    スマホ --> A社
    スマホ --> B社
    PC --> C社
    マウス --> D社
          
```

(関係)モデル

ID	商品名	価格	メ-カ-
001	iPhone	176,000	A01
002	GPixel	98,000	G01
003	Xperia	60,000	S01
:	:	:	:

ID	会社名
A01	Apple
G01	Google
:	:

キー                      バリュー

iPhone	→	(メ-カ-: Apple, 発売年: 2023, 価格: 176000)
GPixel	→	(メ-カ-: Google, 価格: 98000, 在庫: 0)
Xperia	→	(メ-カ-: SONY, 発売年: 2022)

{ \_\_\_\_\_ : 行を一意に特定できる列

  \_\_\_\_\_ : 他の表を参照するための列

★ \_\_\_\_\_ は \_\_\_\_\_ で操作できる

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① データベース内のデータの表現形式を(① \_\_\_\_\_ )という。行と列で構成される表形式でデータを格納する(①)を(② \_\_\_\_\_ )という。行を一意に特定できる列を(③ \_\_\_\_\_ )と呼び、他の表を参照するための列を(④ \_\_\_\_\_ )という。
- ② 近年は構造化されていないビッグデータを扱うために(⑤ \_\_\_\_\_ )と総称されるDBMSが用いられている。

①データベース ②DBMS ③整合性 ④一貫性 ⑤機密性 ⑥可用性



## ▶ ホワイトボード

▶ データの収集と整理

☆ データを活用して問題解決しよう!

① → ② → ③

① データの収集

\_\_\_\_\_ データ：自分で収集

○ \_\_\_\_\_ に合うデータ △ \_\_\_\_\_ や \_\_\_\_\_

\_\_\_\_\_ データ：他人が収集

○ \_\_\_\_\_ に利用 △ \_\_\_\_\_ 情報、 \_\_\_\_\_

② データの整理

\_\_\_\_\_ (データがない)・ \_\_\_\_\_ (測定ミス)・ \_\_\_\_\_ (大きく外れた値) の除去、修正など

[ 尺度水準：データの種類 ]

質的データ	名義尺度	① スマホの機種? 1. iPhone 2. GPixel 3. ...
	順序尺度	
量的データ	間隔尺度	③ 購入年? _____ 年
	比例尺度	④ 1日の使用時間? 動画 _____ 分 SNS _____ 分

[ オープンデータ ]

国や地方自治体、企業などが一般に公開しているデータ。 e-Stat、RESUS など。

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① インターネット上のデータを扱うときは、複数の情報を見て (① \_\_\_\_\_)を確認することが大切である。国などが公開している(② \_\_\_\_\_)を用いると便利。
- ② 収集したデータには値が不明の(③ \_\_\_\_\_)や、測定ミスなどによる(④ \_\_\_\_\_)、他のデータから大きく離れている(⑤ \_\_\_\_\_)が存在する場合があります、修正などの対応が必要である。

①データモデル ②リレーショナルモデル ③主キー ④外部キー ⑤NoSQL



▶ ホワイトボード

▶ データ分析 - 代表値と分散・標準偏差 -

**代表値** データ全体の特徴

①            : (データの合計値) ÷ (データの個数)

	A	B	C	D	E
点数	80	75	90	95	60

(平均値) =

②            (          ) : 大小順で中央の値

	A	B	C	D	E
点数	80	75	0	95	60

(中央値) =



③            (          ) : 最も頻度が多い値

	A	B	C	D	E	F	G	H	I
点数	20	20	20	30	60	80	90	100	100

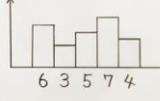
(最頻値) =

**分散と標準偏差** データのばらつき度合い

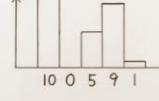
①            :  $\frac{((\text{データ}) - (\text{平均値}))^2 \text{の合計値}}{\text{データの個数}}$

②            :  $\sqrt{\text{①}}$

(1) 漢字



(2) 計算



⑦-⑩  
⑦-⑩)<sup>2</sup>  
分散  
標準偏差

▶ 確認テスト

画面を止めて考えてみよう!

① データ全体の特徴を表す代表値には、次のものがある。

- ・(①           ) : (データの合計値) ÷ (データの個数)
- ・(②           ) : データを大小順に並べたときの中央の値
- ・(③           ) : 最も頻度が多い値

② 5つのデータ[1 2 3 4 5]の(①)は(④   )である。それぞれのデータから(④)を引いて2乗すると[4 1 0 1 4]となるため、分散は(⑤   )、標準偏差は約(⑥           )となる。

①信憑性 ②オープンデータ ③欠損地 ④異常値 ⑤外れ値



## ▶ ホワイトボード

▶ データ分析 - データの可視化 -

★ データを集計してグラフで可視化 → \_\_\_\_\_ や \_\_\_\_\_ を視覚的につかみやすい!

① 時系列:

東京 平均気温

東京 平均降水量

② 割合:

インターネットの用途 (13~19才)

動画	36%
SNS	26%
ゲーム	16%
メール	7%
web	7%
その他	7%

③ 度数分布:

15~19才の学習時間 (過全体・都道府県別)

④ 関連性:

50m走のタイム (16才)

運動時間	~30分	30~60分	60~120分	120分~
男子	7.56	7.44	7.28	7.07
女子	9.17	8.93	8.77	8.34

## ▶ 確認テスト

画面を止めて考えてみよう!

- データを可視化するために、次のようなグラフが用いられる。
- ・(① \_\_\_\_\_): データの分布の様子や特徴を表すグラフで、各階級の幅を横軸、対応する度数を縦軸にとる
  - ・(② \_\_\_\_\_): データの最小値、第1四分位数、中央値、第3四分位数、最大値を視覚的に表す
  - ・(③ \_\_\_\_\_): 2つの変数を横軸と縦軸にして、データを対応する座標の点として表す

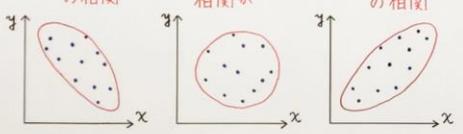
①平均値 ②中央値 ③最頻値 ④3 ⑤2 ⑥1.4



## ▶ ホワイトボード

▶ データ分析 - 相関と回帰分析 -

① 2つのデータ  $x$  と  $y$  の間に  
【 $x$ が(増) →  $y$ が(増) or (減)】の関係があるときに“\_\_\_\_\_ [ \_\_\_\_\_ ] がある”という  
の相関      の相関      の相関

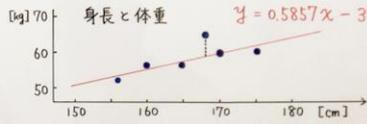


▪ \_\_\_\_\_ : 相関の有無や強弱をみる指標  
-1~1の値をとり、絶対値(大) → 関係性(大)

②  $x$  と  $y$  に相関関係があるとき  
 $x$  と  $y$  の関係を式で表せる( \_\_\_\_\_ )

- \_\_\_\_\_ : 関係式を求めること
- \_\_\_\_\_ :  $y = ax + b$  の形

身長と体重  $y = 0.5857x - 39.027$



▪ \_\_\_\_\_ : (残差)<sup>2</sup>の総和が最小になるように関係式を求める方法

## ▶ 確認テスト

画面を止めて考えてみよう!

- ① 2つの数値データについて、一方の値が増えると他方の値も増減するとき、この2つの値には(① \_\_\_\_\_)があるという。(①)の強さの強弱をみる指標として(② \_\_\_\_\_)があり、-1~1の値で表される。
- ② 2つの値( $x$ と $y$ )に(①)があるとき、 $x$ と $y$ の関係式を求めることを(③ \_\_\_\_\_)という。関係式が $y = ax + b$ の形で表されるとき、この直線を(④ \_\_\_\_\_)とよぶ。

①ヒストグラム ②箱ひげ図 ③散布図



## ▶ ホワイトボード

▶ データ分析 - テキストマイニング -

- \_\_\_\_\_ : 文章(テキスト)のデータを分析して役立つ情報を取り出すこと

🐦 つぶやき 📄 レビュー 📺 コメント

① データの

- 文字情報をコピー
- アンケート回答を入力

YY: 7ヶ月前  
ありがとうございます!  
分かりやすいです。  
Chun: 6ヶ月前  
超絶わかりやすい!

②

- 不要な情報を削除
- 表記ゆれの統一

ありがとうございます!  
分かりやすいです。  
超絶わかりやすい!  
助かりました! 無料で見られるのに感謝

③

① \_\_\_\_\_ : 単語や文節で区切る

この 動画 は わかりやすい です  
連体詞 名詞 助詞 形容詞 助動詞

② データ分析: \_\_\_\_\_ や \_\_\_\_\_ など

名詞	出現頻度
メロス	76
王	19
友	15

いま 笑つて シリアスな アツキ 友 X 王 君 くれる 走る 無いです おまえ

しまつて 悪い 君 無いです おまえ

## ▶ 確認テスト

画面を止めて考えてみよう!

たくさんの文章データを分析して役立つ情報を取り出すことを(① \_\_\_\_\_)という。(①)ではまず分析したいデータを集めて、不要な文字を除去したり表記ゆれを統一したりする前処理を行う。

その後、文章を単語や文節で区切る(② \_\_\_\_\_)をして、各単語の(③ \_\_\_\_\_)や(④ \_\_\_\_\_)などを分析する。

Lesson4-4 の回答 : ①相関 ②相関係数 ③回帰分析 ④回帰直線

Lesson4-5 の回答 : ①テキストマイニング ②形態素解析 ③出現頻度 ④共起語  
(③④は順不同)

# スクショタイムの画像一覧

## ▶ 1章 情報社会の問題解決

### ▶ 情報とは

① なぜ学ぶの？

便利だけど 凶器にもなる！ → 情報技術も同じ → 便利 & 楽しい！ / 周りを助けよう！

① 情報って何？

**情報**  
ある物事や事情の報告で意思決定の判断基準になる

↑ 意味・解釈 ↓ 分析・体系化

**データ**  
観測や調査などで得られる客観的事実

**知識**  
情報が役立つ形で蓄積されたもの

② 情報の特性

モノとしての形がない

① 残存性：伝えてもなくなるらない  
② 複製性：簡単にコピーできる  
③ 伝播性：瞬時に伝わる

情報とは

### ▶ メディアの特性

① **メディア**：情報を伝える手段や媒体。情報を **表現 / 伝達 / 記録** するものに分類

① **表現メディア**：文字、画像、音声、動画

② **伝達メディア**：手紙、本、新聞、電話、テレビ、ラジオ

③ **記録メディア**：手帳、ハードディスク、USBメモリ、フラウドストレージ

正確に伝わる？ 使える？ 重要？ 状況？ 伝わる人の範囲？ 目的？ 保管期間？ 情報量？ 他人に渡す？

メディアの特性

### ▶ 問題解決 - 発想法のツール

① 問題の発見：現実と理想のギャップに気付く

② 問題の明確化：問題を具体化してゴールを設定

③ 解決案の検討：情報を収集・分析して解決案を考える

④ 計画と実行：計画を立てて実行する

⑤ 振り返り：結果を評価して次に活かす

① **ブレインストーミング**  
複数の人で多くのアイデアを出す手法  
◎ 質より量 ◎ 自由でユニーク ◎ 他人に便乗 × 否定 & 批判

② **KJ法**  
カードでアイデアを整理する手法  
【書く → グループ化 → 図示】

③ **マインドマップ**  
中心から枝を伸ばして発想を広げ整理する手法

発想法のツール

### ▶ 問題解決 - 計画管理のツール

① 問題の発見：現実と理想のギャップに気付く

② 問題の明確化：問題を具体化してゴールを設定

③ 解決案の検討：情報を収集・分析して解決案を考える

④ 計画と実行：計画を立てて実行する

⑤ 振り返り：結果を評価して次に活かす

① **PERT図**  
作業間の順序関係を表現

② **ガントチャート**  
作業内容と期間の関係を表現

計画管理のツール

### ▶ 知的財産権

① **知的財産権**：知的な創作活動から生まれた価値を保護

① **産業財産権**：ビジネス → 特許庁への出願で発生  
・ 特許権 [発明・20年] [特許法・25年]  
・ 実用新案権 [アイデア・10年] [特許法・10年]

② **著作権**：文化・芸術 → 創作時に発生

他人の著作物を利用するとき

原則 作者の 許諾 を得る  
例外 保護期間(70年)を経過 私的利用のコピー 教育機関でのコピー 非常利目的の上演 など

引用のルール  
① 必要性がある ② 全体の一部 ③ 他と 区別 可 ④ 出典 の明記

知的財産権

### ▶ 個人情報

個人を識別できる情報

基本4情報：氏名、性別、生年月日、住所

流出：情報漏洩(攻撃、紛失)、メアド、学歴、職業 など

個人情報保護法

個人情報取扱事業者の義務

情報セキュリティポリシー

基本方針：価値観、対策基準：ルール、実施手順：具体的方法

プライバシーポリシー

流出された人：悪戯解雇、損害賠償、窃盗罪 など  
流出された人：売買、DM、不正送金、迷惑メール など

個人情報

### ▶ サイバー犯罪

① **サイバー犯罪**：コンピュータやネットワークを悪用した犯罪

① **フィッシング詐欺**  
① メール受信/広告表示  
② 偽リンクをクリック  
③ ID/パスワード等入力

② **ワンクリック詐欺**  
① web閲覧/メール受信  
② リンクをクリック  
③ 契約成立の画面表示

③ **ショッピング詐欺**  
① ショッピングサイト閲覧  
② 注文・送金  
③ 商品が届く/届かない

不正アクセス禁止法違反  
ID/パスワードを不正入手・使用

コンピュータ電磁的記録対象犯罪  
コンピュータの不正利用やデータの改ざん・破壊

ネットワーク利用犯罪  
詐欺・誹謗中傷 など

対策  
★ 迷惑メールフィルター  
★ テキスト形式で受信  
★ ログイン通知

★ 注文前に確認!  
★ 運営者情報  
★ 決済方法  
★ 価格 ④ 文面

サイバー犯罪

### ▶ 不正ソフトウェア

① **マルウェア**：被害を与えようという悪意のあるソフトウェアの総称

① **コンピュータウイルス**  
① 自己伝染 ② 潜伏 ③ 発病  
・ データ破壊、情報漏洩、遠隔操作 など

② **トロイの木馬**  
偽装アプリとして侵入  
・ 情報漏洩、遠隔操作

③ **ワーム**  
ネットワークで自己増殖  
・ 高負荷、動作停止

④ **ランサムウェア**  
データ暗号化・機能制限  
・ 金銭の要求

⑤ **スパイウェア**  
隠れて情報収集・送信  
・ 情報漏洩

流入  
動作が違い、突然再起動

感染  
① ネットワーク切断  
② 状況確認  
③ マルウェア駆除

① メッセージツール  
② web閲覧  
③ ファイル共有  
④ 記録メディアなど

① ウィルス対策ソフトウェア  
② 定義ファイルの更新  
③ OS等の 定期的アップデート  
④ アプリ 最低限 の確認  
⑤ 定期的にバックアップ

不正なソフトウェア

### 情報モラル

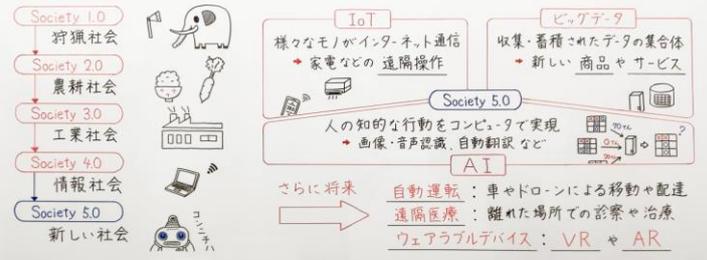
適切な **情報モラル** (情報社会に対する行動や考え方) をもって楽しく便利に!

- ① すべての通信は **往復ハガキ**
- ② **あなた好み** の情報に囲まれる
- ③ 正義感の **デマ** や **チェーンメール**
- ① **知的財産権** や **個人情報** は大丈夫?
- ② **目の前** にいても **その言葉で話せる?**
- ③ 発信内容への **リアクション**

情報機器の使いすぎによる **VDT障害** やスマホ/ネット **依存性** に注意!

## 情報モラル

### 情報技術の発展



## 情報技術の発展

## 2章 コミュニケーションと情報デザイン

### アナログとデジタル

- ① **アナログ**: 連続した量 (**連続量**) で表現すること, analogy (類似) が語源
- デジタル**: とびとびの値 (**離散量**) で表現すること, digit (指) が語源
- ② コンピュータは情報を **0** と **1** で表す
- ③ デジタルデータの特徴

アナログ	デジタル
電圧 高/低	電流 〇/×
磁気 N/S	反射光 向き
デジタル化(A/D変換)	アナログ化(D/A変換)
△ 欠点	△ 欠点
(1) 様々な情報を <b>統合的</b> に扱える!	(1) 細かい情報が <b>失われる</b>
(2) <b>編集</b> や <b>コピー</b> が <b>カンタン!</b>	(2) <b>直観的</b> な理解が <b>難しい</b>
(3) 情報を <b>再現</b> できる!	

## 2-1-1 アナログとデジタル

### 情報量

- ① デジタルでは数値・文字・音声・静止画・動画などを0と1の組合せ (**ビット** **パターン**) で表現できる
- ② 1桁の情報量 = **1ビット** [bit] = 2通り
- ③ **ビット** = **1バイト** [B] = 2<sup>8</sup>(256)通り
- ④ 大きい情報量を表現するのに **接頭語** を使う

3桁は **8** (2<sup>3</sup>) 通り

和

- KB = 1024B
- MB = 1024KB
- GB = 1024MB
- TB = 1024GB

情報量はいくつ?

- (1) 原稿テキスト (2000字) → 7.34 KB
- (2) サムネイル画像 → 2.20 MB
- (3) 動画 (5分) → 1.06 GB

## 2-1-2 情報量

### 10進数と2進数

10進数	2進数
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

① **10進数**: 10になると桁が進む数

② **2進数**: 2になると桁が進む数

【2<sup>8</sup>=256, 2<sup>7</sup>=128 は覚えろと便利!】

## 2-1-3 10進数と2進数

### 16進数

① 2進数は **桁** が多くなる

② 2進数は **桁** が多くなる

③ 2進数は **桁** が多くなる

④ 2進数は **桁** が多くなる

⑤ 2進数は **桁** が多くなる

## 2-1-4 16進数

### 2進数の加減算と補数

① 2進数の加減算

② **補数**: その数を基準となる数から引いた数

③ **補数**を使った減算

④ 2の補数を足す (+10000-0011)

⑤ 繰り上りを切捨て (-10000)

## 2-1-5 2進数の加減算と補数

### 2進数の乗除算

① 3×4 + 3を4回 **足す** と? 12÷4 + 12から4を何回 **引ける** ?

② 乗算と除算も **加算** と **反転** だけでOK! さらに **シフト** で効率化!

③ **シフト** (演算): ビットを左右に **ずらす**

## 2-1-6 2進数の乗除算

### ▶ 実数

① **固定小数点数** : 小数点の位置が固定 → 絶対値が大きい(細い)値を表現できない

10進 123.45 (10) →  $1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$   
 2進 101.01 (2) →  $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 5.25 (10)$

② **浮動小数点数** : 小数点の位置が移動 → **正規化**してよく使われる

10進 6.02 × 10<sup>23</sup>  
 符号 正 仮数 基数  
 ③ 123.45  
 ④ 1.2345 × 10<sup>2</sup>

2進 101.01 × 2<sup>0</sup>  
 10.101 × 2<sup>1</sup>  
 1.0101 × 2<sup>2</sup>  
 0.10101 × 2<sup>3</sup>  
 0.010101 × 2<sup>4</sup>

1.0101 × 2<sup>23</sup>

符号部 (+0-+1)  
 指数部 11(8) 1023(17)を足す  
 仮数部 011010100  
 整数部分の1は省略

## 2-1-7 実数の表現方法

### ▶ 誤差

① **丸め誤差** : 下位の桁を切り捨て、切り上げ、四捨五入(丸め)などで発生

$0.2 = 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$   
 $0.2 = 0.00110011$  (切り捨て)

→ 桁が大きいデータ型を使う!

② **情報落ち誤差** : 絶対値の小さい値を加減しても結果に反映されない

$(1.0101 \times 2^4) + (1.1111 \times 2^4)$   
 $1.0101000000 \times 2^4$   
 $+ 0.0000011111 \times 2^4$   
 $1.0101$  (反映されない)

→ 絶対値の昇順に計算する!

③ **桁落ち誤差** : 絶対値がほぼ等しい値を減算すると有効桁が減る

$\sqrt{101} = 0.10049875 \times 10^2$   
 $\rightarrow \sqrt{100} = 0.10000000 \times 10^2$   
 $0.00049875 \times 10^2$   
 $\rightarrow 0.49875000 \times 10^1$  (有効桁が(桁数)減る)

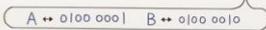
減算しないように式を变形  
 $\sqrt{101} - \sqrt{100} = \frac{(\sqrt{101} - \sqrt{100})(\sqrt{101} + \sqrt{100})}{\sqrt{101} + \sqrt{100}} = \frac{1}{\sqrt{101} + \sqrt{100}}$   
 $\sqrt{101} + \sqrt{100} = 0.20049875 \times 10^2$   
 $1 \div (0.20049875 \times 10^2) = 0.49875623 \times 10^1$

④ **打ち切り誤差** : 無限に続く計算を途中で打切ることによる

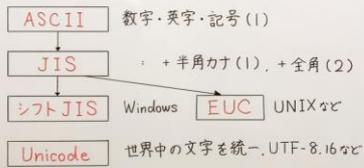
## 2-1-8 誤差

### ▶ 文字のデジタル表現

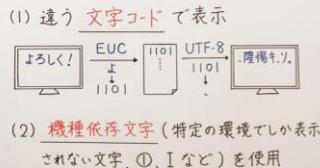
文字にはそれぞれ別のビットパターンが割り当て(文字コード)られている



① 5つの文字コード



② 文字化けの理由



## 2-1-9 文字のデジタル表現

### ▶ 音のデジタル表現

音(空気の振動)は ① 標準化 → ② 量子化 → ③ 符号化 でデジタル化される

PCM (パルス符号変調)方式

電気信号 → ① 標準化(サンプリング) → ② 量子化 → ③ 符号化(コード化)

電圧 時間

精度  
 ① 標準化 周期 : サンプリグの間隔?  
 ② 標準化 周波数 : 1秒間に何回? [Hz] (44.1kHz → 96kHz)  
 ③ 量子化 ビット数 : 何段階に量子化? (16bit → 24bit)

## 2-1-10 音のデジタル表現

### ▶ 色のデジタル表現

① ディスプレイ → 光の三原色 (RGB)  
 - 赤 (Red)・緑 (Green)・青 (Blue)  
 - 赤+緑+青 → 白 : 加法混色 (赤+緑=黄)  
 ② 赤・緑・青の明るさをそれぞれ8ビットで表すと何種類の色を表現できる?  
 赤だけ → 2<sup>8</sup> = 256 種類  
 赤&緑&青 → 256 × 256 × 256 = 16,777,216 種類  
 → 24ビットで約 1677万色を表現 (フルカラー) → 16進数6桁 #FFFFFF

② プリンタ → 色の三原色 (CMY)  
 - シアン (Cyan)・マゼンダ (Magenta)・黄 (Yellow)  
 - 白 + シマ・黄 + 黒 : 減法混色  
 - 実際は CMY + B (黒)  
 [参考] 色の三属性 (HSL)  
 - 色相 (色合い: Hue)  
 - 彩度 (鮮やかさ: Saturation)  
 - 明度 (明るさ: Lightness)

## 2-1-11 色のデジタル表現

### ▶ 画像のデジタル表現

画像は ① 標準化 → ② 量子化 → ③ 符号化 でデジタル化される

① 標準化 : 画像を等間隔のマス目(画素/ピクセル)に区切る  
 ② 量子化 : 色を数値で表現する  
 - 階調 : 色の濃淡を何段階で表現するか  
 - 2値 → 白と黒のみ (1)  
 - ディスプレイ → 横縦の画素数 (1920 × 1080)  
 - プリンタ → 1インチあたりのドット数 (300dpi)  
 - フルカラー → RGB各256段階 (24)

③ 符号化 : 量子化した数値を0と1(2進法)に変換

④ [ペイント] ソフトウェア : 画像を点の集まりで表現 → ラスタ形式  
 [ドロー] ソフトウェア : 画像を座標・角度・太さなどで表現 → ベクタ形式

## 2-1-12 画像のデジタル表現

### ▶ 動画のデジタル化

動画は 静止画像 を連続的に表示したもの (残像現象で動いて見える)

① フレーム : 動画を構成する静止画像  
 フレームレート : 1秒あたりに表示されるフレームの数 [fps]  
 TV : 30fps / 60fps 映画 : 24fps

② フレームレート ⊕ → 動画 ⊙  
 → データ量 △

③ 1920 × 1080ピクセル・24ビットフルカラーのフレーム画像からなる30fps・1分間の動画のデータ量は?  
 画像1枚 : 3[B] × 1920 × 1080 = 6,220,800[B] = 6.075 [KB]  
 動画1分 : 6.075 × 30 × 60 = 10,935,000 [KB] = 10.678 [MB] = 10.43 [GB]

## 2-1-13 動画のデジタル表現

### ▶ データの圧縮

① 圧縮 : データの内容を保ってデータ量を減らす処理 → 展開  
 → データ量 ↓ ・ ファイル名わかる  
 圧縮率 [%] =  $\frac{\text{圧縮後のデータ量}}{\text{圧縮前のデータ量}} \times 100$

② 可逆圧縮 : 圧縮前に戻せる  
 → 文書データ、プログラムなど  
 非可逆圧縮 : 圧縮前に戻せない  
 → 画像、音声、動画など

③ ランレングス圧縮 : データの連続回数に変換  
 ⊠ 白を0, 青を1 → 8 × 8 = 64 [bit]  
 ⊡ データの連続回数を並べると  
 4, 3, 2, 4, 7, 1, 3, 8, 4, 1, 7, 1, 7, 1, 6, 1, 4 [17個]  
 → 3 × 17 = 51 [bit]  
 圧縮率 = (51 ÷ 64) × 100 = 79.7 [%]

④ 画像 : BMP, GIF, PNG, JPEG  
 音声 : WAVE, MP3, AAC, WMA  
 動画 : AVI, MOV, MPEG など

## 2-1-14 データの圧縮

### ▶ コミュニケーションとメディアの変化

- ・ **通信** (コミュニケーション) : 情報をやりとりすること
- ・ **メディア** : 情報をやりとりの方法や媒体. 不特定多数向け → **マスメディア**

技術	15世紀 活版 印刷	19世紀 腕木 通信	19世紀 郵便 制度	電気 通信	20世紀 無線 通信	1969 ARPA NET	1991 インター ネット
個人	口頭	セファール	手紙	電話	携帯電話		スマートフォン
マス	のりし 書籍	新聞		ラジオ	テレビ		双方向テレビ

・ **ソーシャルメディア** : 多数の人々が相互に情報発信するコミュニケーションサービス

## 2-2-1 メディアの発達

### ▶ コミュニケーションとネットリテラシ

① コミュニケーションの形

	同期型	非同期型
1対1	会話 電話 ビデオ通話	LINE メール 手紙
1対多	授業 プレゼン ライブ配信	SNS HP YouTube

○ 反応が分かる  
△ 時間・場所の制約

○ 便利, 簡単しやすい  
△ 勘違い, 消せない

② ネットコミュニケーションの特徴と注意点

- ① **匿名性** : 名前などの情報を隠せる
- ② **信憑性** が保証されていない  
→ 情報を **鵜呑み** にしない!
- ③ **可視性** : 誰にでも見られる可能性
- ④ **持続性** : 発信した情報は消せない  
→ **対面** で言わないことは言わない!
- ⑤ **記録性** : アクセス情報はすべて記録  
→ ネットは **匿名** ではない!

## 2-2-2 メディアの特性

### ▶ 情報デザイン

**情報デザイン** 情報を分かりやすく伝えるための工夫や表現方法

- ① **抽象化** 余分な情報を除いて要点をシンプルに表現  
アイコン, サイン (案内誘導板), **ピクトグラム**
- ② **可視化** 視覚的に分かりやすく表現  
図, 表, グラフ, **インフォグラフィックス** (路線図, ハザードマップ)
- ③ **構造化** 情報を整理して分かりやすく表現  
並列, 順序, 分岐, 階層

## 2-3-1 情報デザイン

### ▶ ユニバーサルデザイン

**ユニバーサルデザイン (UD)** : 年齢や性別, 言語, 身体能力に関係なく皆が使いやすいデザイン

使えない → **アクセシビリティ** 皆が使える? → 使える (使いやすい) → **ユーザビリティ** 皆が使いやすい? → 使いやすい

- ① **情報バリアフリー**  
障がい者や高齢者の障壁を取り除く  
・ 文字の **拡大・読み上げ** 機能
- ② **色覚バリアフリー**  
色覚の多様性を考慮する  
・ 見分けやすい **配色, 色以外** の情報
- ③ **ユーザインタフェース**  
コンピュータの操作環境や情報の表示形式  
→ **CUI・GUI・NUI**
- ④ **ユニバーサルデザインフォント**  
使いやすく読み間違えにくいフォント  
(明朝 → 長文資料, ゴシック → スライド)

## 2-3-2 ユニバーサルデザイン

## ▶ 3章 コンピュータとプログラミング

### ▶ コンピュータの構成

- ・ **ハードウェア** : コンピュータを構成する装置
- ・ **インタフェース** : 装置同士を接続する規格. USB や Wi-Fi, Bluetooth など.

## コンピュータの構成

### ▶ CPUとメモリの仕組み

- ・ **レジスタ**  
CPU内にある記憶領域
- ・ **クロック信号**  
各装置の動作タイミングを合わせる周期的な信号
- ・ **クロック周波数**  
1秒間の信号の回数 (Hz) ↑ 5Hz
- ・ 処理効率を上げる技術  
□ CPUの **ビット数** : 32 → 64  
□ **クロック周波数** を上げる  
→ 国 **マルチコアプロセッサ**

## CPUとメモリの仕組み

### ▶ 演算の仕組み

演算装置は **論理演算** (真と偽の AND, OR など) や **算術演算** (0と1の +, - など) ができる

- ① 基本的な論理回路
  - AND (論理積)
  - OR (論理和)
  - NOT (否定)
- ② **半加算回路** : 2進数1桁の加算回路
- ③ **全加算回路** : 桁上りを考慮した加算回路

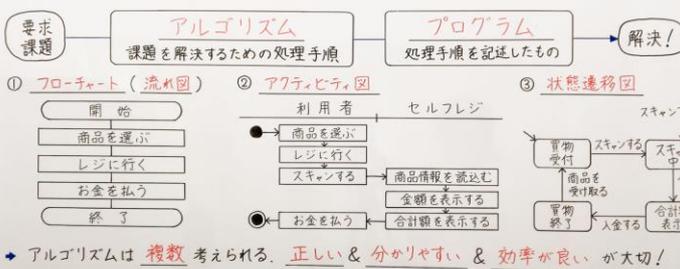
## 演算の仕組み

### ▶ ソフトウェアの種類

- ・ OSの管理機能
  - ① **タスク管理**  
複数の処理 (タスク) に対して CPU を適切に割り当て
  - ② **メモリ管理**  
主記憶装置を効果的に活用
  - ③ **データ管理**  
フォルダ (ディレクトリ) によって ファイルを階層的に管理

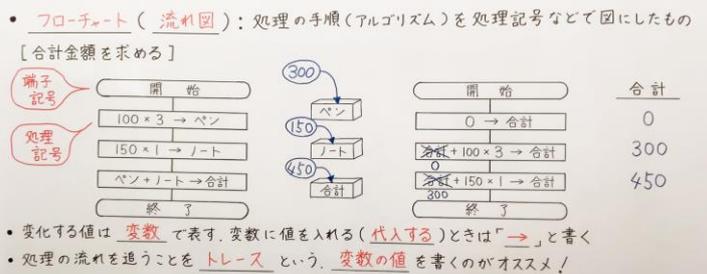
## ソフトウェアの種類

▶ アルゴリズムとは



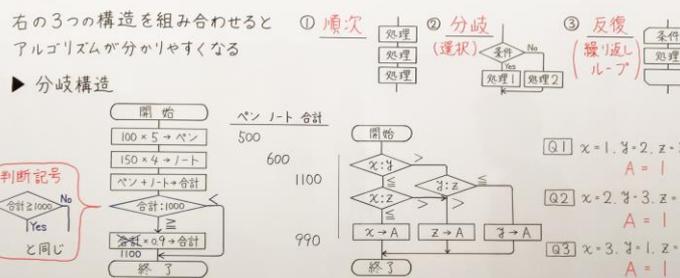
# 1 アルゴリズムとは

▶ フローチャートの基本



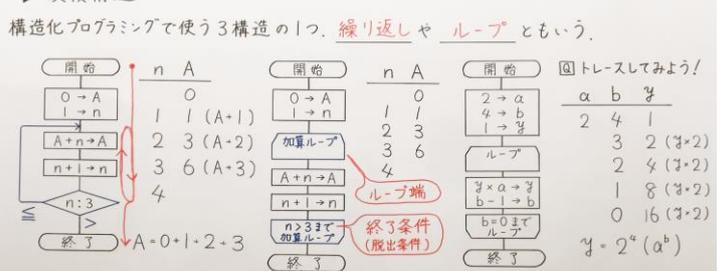
# 2 フローチャートの基本

▶ 構造化プログラミング



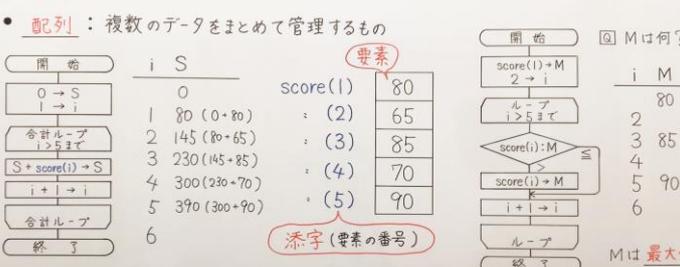
# 3 分岐構造

▶ 反復構造



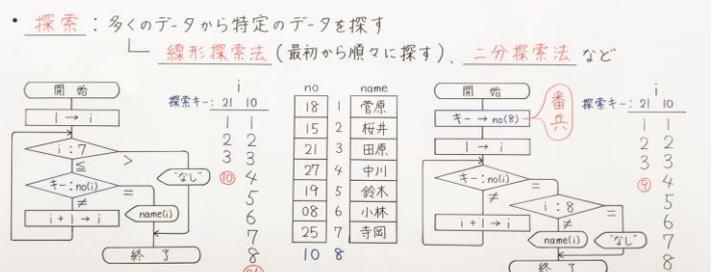
# 4 反復構造

▶ 配列



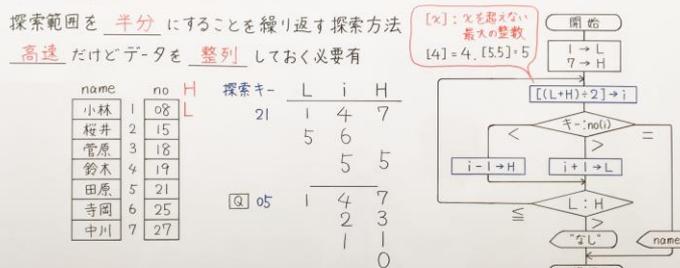
# 5 配列

▶ 線形探索法



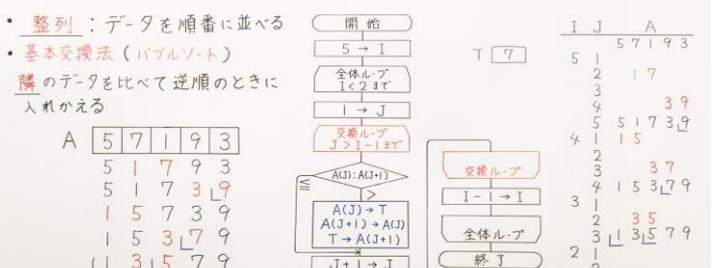
# 6 線形探索法

▶ 二分探索法



# 7 二分探索法

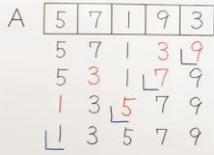
▶ 基本交換法 (バブルソート)



# 8 基本交換法

▶ 基本選択法 (選択ソート)

一番大きい データを選択して  
右端に置く

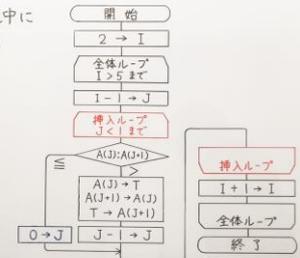
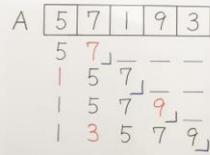


	I	J	M	A
			5	7
5	4	5		
	3	4		
	2	4		
	1	0	5	7
4	3	4		
	2	2		
	1	0	5	3
3	2	3		
	1	2		
	0	1	1	3
2	1	2		
	0	1	1	3
1	0	1		

9 基本選択法

▶ 基本挿入法 (挿入ソート)

先頭 から整列しながら、途中に  
入るデータがあれば挿入する

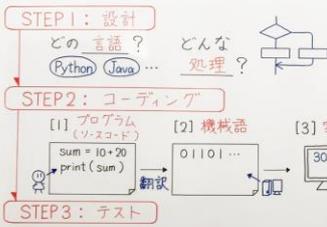


	I	J	A
			5
2	1		
	0	5	7
3	1	7	
	1	1	5
0	1	5	7
4	0	1	5
	0	1	5
5	4	0	1
	3	3	7
2	1	0	1
	0	1	3
1	0	1	3
	0	1	3

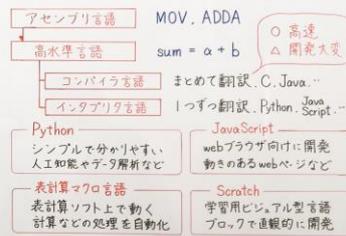
10 基本挿入法

▶ プログラミング

① プログラミングの流れ



② プログラミング言語の種類



プログラミング

▶ 表計算マクロ言語とは

① 表計算 ソフトウェアの マクロ 機能を実現するプログラミング 言語



大きな処理 (複数の処理をまとめたもの) を自動化  
【例】 データチェック → 集計 → 書類作成

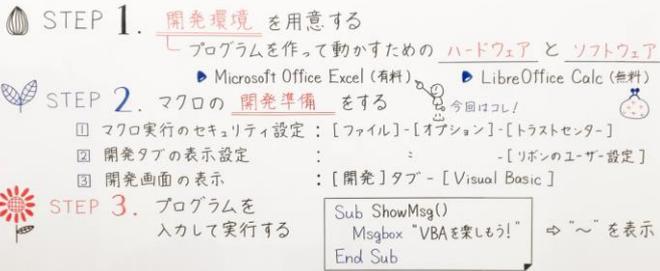
② VBA (Visual Basic for Applications)

1993 年に Microsoft 社が公開した Office シリーズで動くプログラミング言語  
→ 作業の自動化、アプリ間の連携、OS の操作

- 特徴 ① 特別な プログラミング環境 が不要!
- ② プログラムが シンプル かつ 動かしやすい!
- ③ セキュリティ に要注意! → 信頼 できるマクロだけ実行しよう

01 表計算マクロ言語とは

▶ VBA マクロを動かそう

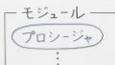


02 VBAを動かそう

▶ VBAの基本

① プログラムの構造

```
Sub プロシージャ名()
    (処理)
End Sub
```



③ セルの指定方法

Cells(行,列).Value	A	B
Range("セル範囲").Value	1	
Cells(3,2).Value = 10	2	
	3	10

② 基本的なルール

- 処理は 改行 で区切る
- 処理のまとまりを インデント (下げ) で指定する
- 文字列は " " で囲む
- コメント ( ' ) でメモ書きを入れる

④ 実行ボタンの作成

```
Sub WriteCell()
    Cells(1,1).Value = "英語"
    Range("B1").Value = 90
End Sub
```

03 VBAの基本

▶ 変数

① 変数はデータの 入れ物



② 変数名のルール

- 英数字、ひらがな、カタカナ、漢字、\_(アンダースコア) を使える
- 大文字・小文字は区別 しない
- name, student\_no
- |文字目に 数字・記号 は使えない
- キーワード(予約語) は使えない
- !st\_number, sub

③ 変数の使い方

- 宣言 する: 変数を作る  
Dim 変数名(, 変数名, ...)  
Dim age
- 代入 する: 値を入れる  
変数名 = 値  
age = 16
- 参照 する: 値を見る  
変数名 (→ 値に置きかわる)  
Range("B1").Value = age



04 変数

▶ データ型

① データ型: データの種類

- 整数 (10, 365)
- 小数 (3.1415...)
- 文字列 ("VBA")
- 真偽値 (True, False)

② 変数の宣言

```
Dim 変数名 As データ型
Dim age As Byte
age = 16
age = 256 → エラー!
```

		0000 0000 ~ 1111 1111
整数	Byte	8ビット 0 ~ 255
	Integer	16 -32768 ~ 32767
小数	Long	32 -2147483648 ~ 2147483647
	Single	32 単精度 浮動小数点数
文字列	Double	64 倍精度
	String	80+文字列長 約 65,400 文字
真偽値	Boolean	16 True, False のみ
	Date	64 100年11月1日 ~ 9999年12月31日
バリエーション	Variant	128 [数値] Double と同じ [文字列] String と同じ

05 データ型

## 配列

- 配列: 複数の変数をまとめて管理
- 配列の使い方
  - 宣言する (添字(0から))
 

```
Dim 配列名(要素数-1) As テータ型
                    ' (最小値 To 最大値)
                    Dim score(0 To 2) As Long
                    Dim score(2) As Long
                    Dim score(0 To 2) As Long
```
  - 代入する
 

```
配列名(添字) = 値
                    score(0) = 80
```
  - 参照する
 

```
配列名(添字)
                    score(0) → 80
```
- 便利な関数 (処理をまとめたもの)
  - Array(要素, 要素, ...): 配列の作成
 

```
Dim name As Variant
                    name = Array("菅原", "梅井", "田原")
```
  - LBound(配列名): 最小の要素番号
  - UBound(配列名): 最大の要素番号
 

```
LBound(name) → 0
                    UBound(name) → 2
```

## 06 配列

## 演算子

$7 = 2 \times 3^2 - 4$

先 ↑	^	累乗
①	* /	乗除
	\(¥) Mod	商余り
	+ -	加減
②	&	連結
③	< > <= >=	大小比較
	= <>	等しいか
④	Not	～ない
	And	かつ
後 ↓	Or	または

- 算術演算子
  - $7^2 \rightarrow 49$      $7 / 2 \rightarrow 3.5$
  - $7 \setminus 2 \rightarrow 3$      $7 \text{ Mod } 2 \rightarrow 1$
- 文字列連結演算子
  - $7 \& 2 \rightarrow 72$
- 比較演算子
  - $a > 0 \rightarrow \alpha$  が 0 以上なら True
- 論理演算子
  - $(a > 0) \text{ And } (b > 0)$ 
    - 結果は必ず True か False
    - $\rightarrow \alpha$  が 0 以上かつ  $b$  が 0 以上なら True

## 07 演算子

## 分岐処理

- If...Then...Else ステートメント [If文]
 

```
If 条件式 Then
    処理 1
Else
    処理 2
End If
```
  - ElseIf ステートメント [ElseIf節]
 

```
If 条件式 A Then
    処理 1
ElseIf 条件式 B Then
    処理 2
Else
    処理 3
End If
```
- 処理は複数行記述できる
  - どちらかの処理しか実行されない
  - Else ステートメントは省略できる
  - ElseIf ステートメントは複数記述できる
  - 条件式は上から順に判定

## 08 分岐処理

## 反復(繰り返し)処理

- Do...Loop ステートメント [While文]
 

```
Do While 条件式
    処理
Loop
```
  - For...Next ステートメント [For文]
 

```
For 変数名 = 開始値 To 終了値
    処理
Next 変数名
```
- 代表的なライブラリ
- |   |   |   |   |
|---|---|---|---|
| i | A | B | j |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 |
- For j = 1 To 3  
Cells(j, 2).Value = j  
Next j

## 09 反復処理

## 関数

- 関数: 特定の処理をまとめたもの
- Sub プロシージャ << 戻り値なし >>
 

```
Sub プロシージャ名(引数, 引数, ...)
```
- Function プロシージャ << 戻り値あり >>
 

```
Function プロシージャ名(引数, 引数, ...)
```

引数: 呼び出し元から受け取る値  
戻り値: 呼び出し元に返す値  
返す値はプロシージャ名に代入

```
Call ShowMsg()
Call Calc(90, 80)
Function Judge(e, m)
    If (e >= 80) Then
        Judge = "合格!"
    Else
        Judge = "不合格"
    End If
End Function
status = Judge(90, 80)
```

## 10 関数

## ライブラリ

- ライブラリ: よく使う処理を利用しやすくまとめたもの (開発画面で) [ツール]-[参照設定]で指定
- VBA関数: VBAで提供されている関数
  - MsgBox("文字列"): ダイアログボックスに文字列を表示
  - Array(値, 値, ...): 引数を要素にもつ配列を取得
  - Rnd(): 0以上1未満の乱数を取得
  - Int(数値): 数値の整数部分を取得
- API (Application Programming Interface): プログラムやサービスの機能を外部から使う仕組みや利用方法

## 11 ライブラリ

## モデル化とシミュレーション

問題発生 → モデル化: 対象を単純化して模型や数式, 図で表現 → シミュレーション: モデルを使って試行 → 解決

手順

- 問題の明確化
- 構成要素の分析
- 数式や図で表現
- シミュレーション
- モデルを修正
- 問題を解決

表現方法? (物理モデル, 論理モデル)

特性? (動的モデル, 静的モデル, 確率モデル, 確定モデル, 連続モデル, 離散モデル)

時間変化? 不確定要素? 連続データ?

## モデル化とシミュレーション

## シミュレーション①: つり銭問題

STEP 1 問題の明確化  
文化祭で200円の商品 → 何枚必要?

STEP 2 構成要素の分析  
100の枚数, 500の枚数, 割合, 未客数

STEP 3 数式や図で表現  
200円/500円は50% 未客は50名  
支払い(200円/500円)の割合, 未客数  
乱数が 0.0~0.5 → 200円 → +2枚  
0.5~1.0 → 500円 → -3枚

STEP 4 シミュレーション

```
coin = 0
For i = 1 To 50
    rn = Rnd()
    If (rn < 0.5) Then
        coin = coin + 2
    Else
        coin = coin - 3
    End If
    Cells(i, 1).Value = coin & "枚"
Next i
```

5) モデルを修正 6) 問題を解決  
支払いの割合 おつりを用意  
1000円札は? 紙幣はお断り

## 12 つり銭問題

▶ シミュレーション②：待ち行列

**STEP 1** 問題の明確化  
 バイト待ち行列 → 待ち時間？ 短縮時間？

**STEP 2** 構成要素の分析  
 レジ (1台), サービス時間 (30秒)  
 来客数 (100人), 到着間隔 (0~99秒)

**STEP 3** 数式や図で表現

□ 待ちなし

finish ≤ arrival  
 → start = arrival

□ 待ちあり

start - arrival  
 finish > arrival  
 → start = finish

**STEP 4** シミュレーション

```

start = 0 | arrival = 0
finish = 0 | wait = 0
For i = 1 To 100
  rn = Int(Rnd() * 100)
  arrival = arrival + rn
  If (finish < arrival) Then
    start = arrival
  Else
    start = finish
  End If
  wait = start - arrival
  finish = start + 30
  (各変数の値をセルに出力)
Next i
    
```

5) モデルを修正 6) 問題を解決  
 ・ サービス時間を **短縮** ・ レジを **増やす**

13 待ち行列

▶ 4章 情報通信ネットワークとデータの活用

▶ ネットワークの基本

- ネットワーク (NW): コンピュータなどの機器をつないで情報をやり取りする通信網
- ① LAN: 限られた範囲内のNW
  - 有線LAN, 無線LAN
- ② WAN: 遠隔地をつなぐNW
  - 回線交換: 回線を占有
  - パケット交換: 回線を共有
- ③ インターネット: 世界中のコンピュータやNWをつないだNW. ISPと契約必要
- NWの構成要素: NW機器 (ハードウェア), アプリ (ソフトウェア), 通信ルール (プロトコル)

ネットワークの基本

▶ 有線LAN (イーサネット)

- 有線LAN: 限られた範囲内でケーブルをつないで通信するネットワーク → 無線より **安定 & 高速**  
 主流の規格(ケーブルや通信方法の標準ルール)は **イーサネット**

[ケーブル]

- ① **同軸** ケーブル: 銅線を絶縁体などで覆う → Δ 太くて不便
- ② **ツイストペア** ケーブル (= LAN ケーブル): 8本の銅線を2本ずつより合わせる → ○ ノイズに強い, 細くて柔軟, 安価
- ③ **光ファイバー** ケーブル: ガラスやプラスチックで光信号を送る → ○ 高速 Δ 破損, 高価

[機器]

- ① **ルータ**: LANを他のネットワークに接続
- ② **集線装置**: 機器同士を接続
  - ハブ (リビタハブ): 全ポートに送信
  - スイッチ (スイッチングハブ): 宛先に送信

有線LAN

▶ 無線LAN

- 無線LAN: 限られた範囲内で電波を使って通信するネットワーク  
 → データを盗み見されないように **認証・暗号化** でセキュリティを確保!

[無線通信の技術]

	Wi-Fi	Bluetooth
規格	IEEE 802.11 (50~100m, ~数Gbps) → 遠距離・高速	IEEE 802.15.1 (数~10m, ~24Mbps) → 省電力・接続易
用途	アクセスポイント	Bluetooth

[無線LANの規格]

名前	略称	周波数 [GHz]	速度 [bps]
IEEE 802.11a	x	5	54M
b	x	2.4	11M
g	x	2.4	54M
n	Wi-Fi 4	2.4/5	600M
ac	5	5	6.9G
ax	6	2.4/5	9.6G

無線LAN

▶ TCP/IP 4階層モデル

- プロトコル (通信規約): 通信の手順やデータ形式などに関する約束ごと, **階層化** してまとめられている
- TCP/IP 4階層モデル: インターネットの通信をモデル化

階層	機能	プロトコル	表現
4層 アプリケーション層	アプリケーション間の通信ルール → アプリケーション同士でやり取り	HTTP, POP, SMTP, FTP	友達, 家族, 先生 日本語, 英語 電報, 手紙, メール
3層 トランスポート層	エラー検出・訂正などの通信制御 → データが正確に届く	TCP, UDP	成功/失敗, 文字コードなど データの順番, 正しいの判別情報
2層 インターネット層	データの転送経路を選択 → 正しい宛先にデータが届く	IP	住所 宛先・送信元のアドレス情報
1層 ネットワーク層	物理的に接続 → 隣接する機器にデータが届く	イーサネット	住所 NW機器や信号などの情報

TCP/IP 4階層モデル

▶ IPアドレス

- IPアドレス: TCP/IPネットワークで各コンピュータを識別するための固有のアドレス番号

IPv4: 32ビット → 2<sup>32</sup> (約43億)通り

- ① 8ビットずつで区切って **10** 進数表記
- ② ネットワークを識別する **ネットワーク** 部と各機器を識別する **ホスト** 部で構成
- ③ ホスト部すべて0 → **ネットワーク** を表現  
 すべて1 → **すべての** 機器を表現

・ **グローバル IPアドレス**: 世界で唯一  
**プライベート**: LAN内で使用

IPv6: 128ビット → 2<sup>128</sup> (約360億)通り

11000000	10101000	00000000	00001001
192	168	0	9
クラスA	0   7ビット	24ビット (約1,600万)	0 ~ 255
B	10   14ビット	16ビット (65,534)	
C	110   21ビット	8ビット (254)	

ネットワーク部      ホスト部

IPアドレス

▶ ルーティング

- ルーティング (経路制御): IPの役割の1つ, 宛先までの最適な経路を判断してデータが届ける  
 経路は各機器の **ルーティングテーブル** (経路制御表) をもとに決められる

[ルータ①のルーティングテーブル]

宛先	ゲートウェイ	インタフェース	メトリック
192.168.1.0	直接	192.168.1.1	1
192.168.3.0	直接	192.168.3.1	1
192.168.2.0	192.168.3.2	192.168.3.1	2

[ルータ②のルーティングテーブル]

宛先	ゲートウェイ	インタフェース	メトリック
192.168.2.0	直接	192.168.2.1	1
192.168.3.0	直接	192.168.3.2	1
192.168.1.0	192.168.3.1	192.168.3.2	2

ルーティング

### 通信の信頼性

- ★ 通信時はデータのビット **誤り** や **消失** が発生する可能性有  
→ 受信側で誤りを **検出** して、**訂正** や **再送** 依頼の仕組みが大切!
- ① **パリティチェック** (奇偶検査)  
「1」の個数を調べて誤りを **検出**  

データ	パリティビット	受信後
1001100	0	10011000
1001100	1	10011001

  - 奇数パリティ → 0 を付加
  - 偶数 → 1 を付加
- ② **TCPの通信制御**  
データの **順序** を管理して  
不連続時は **再送** を依頼
- ★ **チェックディジット** : データの誤り検出のために元データに付加される数字。バーコードやマイナンバーなどで活用。

## 通信の信頼性

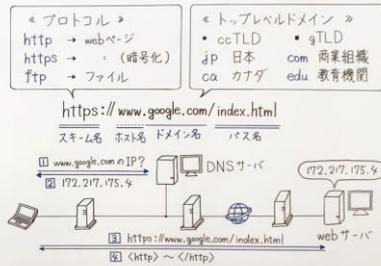
### Webの仕組み

- ① **WWW** (World Wide Web) : インターネット上のwebページを閲覧できるサービス
- ② **webブラウザ** : webページを閲覧するソフトウェア
- ③ **webサーバ** : webページを管理するコンピュータ
- ④ **HTML** : webページを作るための言語  
ハイパーテキスト : 他の文書とリンク  
マークアップ言語 : タグで構造化
- ⑤ **URL** : 情報の場所を指定する記述形式  
例: https://google.com/送信元IPアドレス入力情報 など
- ⑥ **クライアントサーバシステム** : サービスの依頼者(クライアント)と提供者(サーバ)で構成されるシステム  
例: 宛先IPアドレス <html> ~ </html>

## Webの仕組み

### URLとDNSの仕組み

- ① **URL** : 情報の場所を指定
- ② **ドメイン名** : インターネット上のコンピュータやネットワークを識別する名前 → **URL** や **メールアドレス** で使用  
★ 実は **IPアドレス** を変換したもの
- ③ **DNS** : ドメイン名とIPアドレスを対応つけて変換 (**名前解決**) する仕組み



## URLとDNSの仕組み

### 情報セキュリティ

- (情報)セキュリティ : ① **許可** された人だけが ② **正確** な情報を ③ **必要** なときに ④ **必要** なようにすること
- ① **機密性 (Confidentiality)** : 許可された人だけが情報にアクセスできる  
★ ユーザ認証, 暗号化
- ② **完全性 (Integrity)** : 情報が壊れたり変更されたりしない  
★ デジタル署名
- ③ **可用性 (Availability)** : 必要ときに情報を利用できる  
★ システム障害, 複製, 冗長化

## 情報セキュリティ

### 認証とアクセス制御

- 情報の **機密性** (許可された人だけがアクセス可) を高める技術 → **ユーザ認証**, **アクセス制御** など
- ① **ユーザ認証** : ユーザーが正規の利用者かを確認  
★ ユーザID & パスワード + 多要素認証  
  - PIN : 個人識別番号
  - SMS 認証 : 携帯宛てコード
  - ICカード
  - 生体認証
- ② **アクセス制御** : 特定の利用者だけがデータを扱えるように制限  
★ **ファイアウォール** : ネットワークの出入口に設置する装置 or ソフトウェア → **ヘッダ** をもみて **通過** の判断

## 認証とアクセス制御

### 暗号化の仕組み

- 暗号化** : 情報を送るときに、盗聴されても解読できない形式に変換すること
- ① **共通鍵暗号方式** : 送(変) 同じ鍵 (秘密鍵) を使用
- ② **公開鍵暗号方式** : 送(送) **公開鍵** で暗号化 受(受) **秘密鍵** で復号

## 暗号化の仕組み

### デジタル署名と電子認証

- ★ **デジタル署名** : 公開鍵暗号の技術を使ってデータの改ざんがないことを証明
- ① **ハッシュ関数** で文書から **要約文** を取得
- ② ①を **秘密鍵** で暗号化 → **デジタル署名**
- ③ 文書、②、**電子証明書** を送付
- ④ ①と同じ
- ⑤ デジタル署名を **公開鍵** で復号 → 同じ?
- ★ **電子認証** : 認証局が発行する電子証明書でデジタル署名が本人のものであることを証明
- ★ **SSL/TLS** : 暗号化や電子認証を用いる通信技術。URLが https://~となる。

## デジタル署名と電子認証

### 情報システム

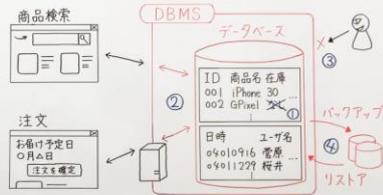
- 情報システム** : 情報機器をネットワークでつないで全体で情報処理をする仕組み [身近な例]
- ★ **ATM** : 入金や出金、振込などができる
- ★ **GPS** : 人工衛星を使って地球上の位置を特定するシステム
- ★ **ETC** : 高速道路の料金支払を自動化するシステム
- ★ **POS** : 商品の販売数や購買情報を管理するシステム

## 情報システム

## データベースの基本

- データベース (DB) : ある目的のためにデータを整理・蓄積したもの
- データベース管理システム (DBMS) : データベースを管理するソフトウェア

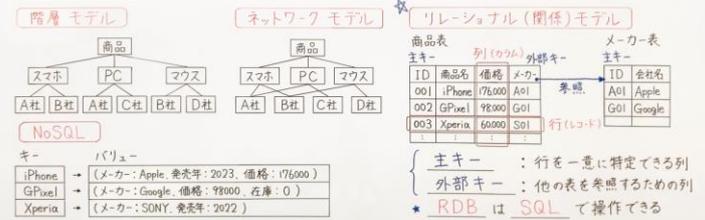
- 整合性 : データの重複や、不正なデータの登録や更新を防ぐ
- 一貫性 : 複数のユーザが同時に操作しても矛盾が生じないようにする
- 機密性 : ユーザ認証をしたりアクセス権を設定したりできる
- 可用性 : 障害に備えてバックアップしたり障害時にリストアしたりできる



## データベースの基本

## データモデル

- データモデル : データベース内のデータの表現形式



## データモデル

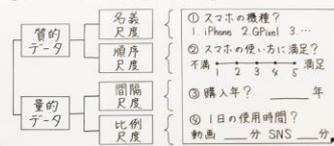
## データの収集と整理

★ データを活用して問題解決しよう!

- ① 収集 → ② 整理 → ③ 分析

- データの収集
  - 一次データ : 自分で収集
    - 目的に合うデータ △ 時間 や 費用
  - 二次データ : 他人が収集
    - 簡単に利用 △ 古い 情報、信憑性
- データの整理
  - 欠損値 (データがない) ・ 異常値 (測定ミス) ・ 外れ値 (大きく外れた値) の除去、修正など

[ 尺度水準 : データの種類 ]



[ オープンデータ ]

国や地方自治体、企業などが一般に公開しているデータ。e-Stat, RESUS など。

## データの収集と整理

## データ分析 - 代表値と分散・標準偏差 -

代表値 データ全体の特徴

- 平均値 : (データの合計値) ÷ (データの個数)
 

A	B	C	D	E
80	75	90	95	60

$$\text{平均値} = \frac{(80+75+90+95+60)}{5} = \frac{400}{5} = 80$$
- 中央値 (メジアン) : 大小順で中央の値
 

A	B	C	D	E
80	75	90	95	60

$$\text{中央値} = 75$$
- 最頻値 (モード) : 最も頻度が多い値
 

A	B	C	D	E	F	G	H	I
20	20	20	30	60	80	90	100	100

$$\text{最頻値} = 90$$

分散と標準偏差 データのばらつき具合

- 分散 :  $\frac{(\text{データ} - \text{平均値})^2 \text{の合計値}}{\text{データの個数}}$
- 標準偏差 :  $\sqrt{\text{分散}}$ 
  - 漢字
  - 計算

## データの可視化

## データ分析 - データの可視化 -

★ データを集計してグラフで可視化 → 傾向 や 特徴 を視覚的につかみやすい!

- 時系列 : 折れ線グラフ, 棒グラフ
  - 東京 平均気温 (折れ線)
  - 東京 平均降水量 (棒)
- 割合 : 棒グラフ, 円グラフ
  - インターネットの用途 (13~19才) (円)
- 度数分布 : ヒストグラム, 箱ひげ図
  - 15~19才の学習時間 (棒)
  - 50m走のタイム (箱ひげ)
- 関連性 : クロス集計表, 散布図
  - 運動時間 vs 50m走のタイム (散布)

## データの可視化

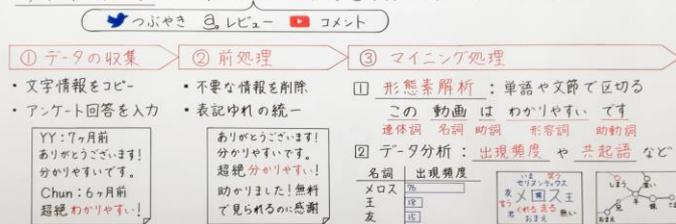
## データ分析 - 相関と回帰分析 -

- 2つのデータ X と Y の間に [ X が増 → Y が (増 or 減) ] の関係があるときに "相関 [ 関係 ] がある" という
  - 負の相関
  - 相関がない
  - 正の相関
- X と Y に相関関係があるとき X と Y の関係を式で表せる (回帰)
  - 回帰分析 : 関係式を求めること
  - 単回帰分析 :  $y = ax + b$  の形
    - 身長と体重  $y = 0.5837x - 39.027$
    - 残差
    - 回帰直線
  - 最小二乗法 : (残差)<sup>2</sup>の総和が最小になるように関係式を求める方法

## 相関と回帰分析

## データ分析 - テキストマイニング

- テキストマイニング : 文章 (テキスト) のデータを分析して役立つ情報を取り出すこと



## テキストマイニング