

Class 7 -4

コンピュータの基本操作

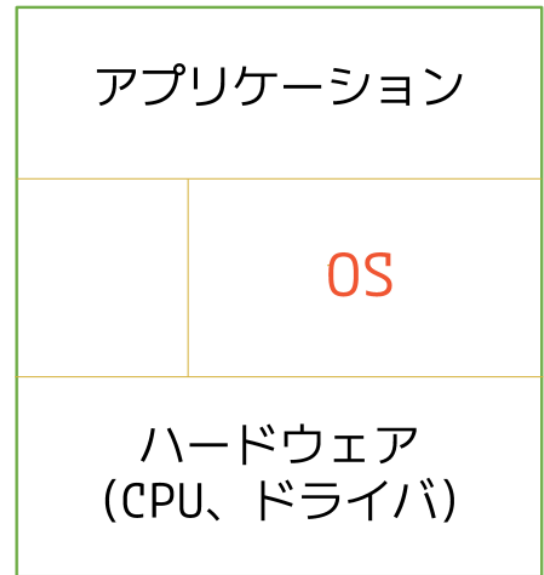
問題解説資料

オペレーティングシステム

- オペレーティングシステム

[Operating System, OS]とは、他のアプリケーションを管理・実行するために使用される**ソフトウェア**の一種です。

- 同時に、オペレーティングシステムは、ソフトウェアとハードウェアの橋渡しをする役割を担っています。OSがあれば、私たちソフトウェア技術者は、特定のハードウェアの機能と直接やりとりすることなく、より効率的に開発を進めることができます。

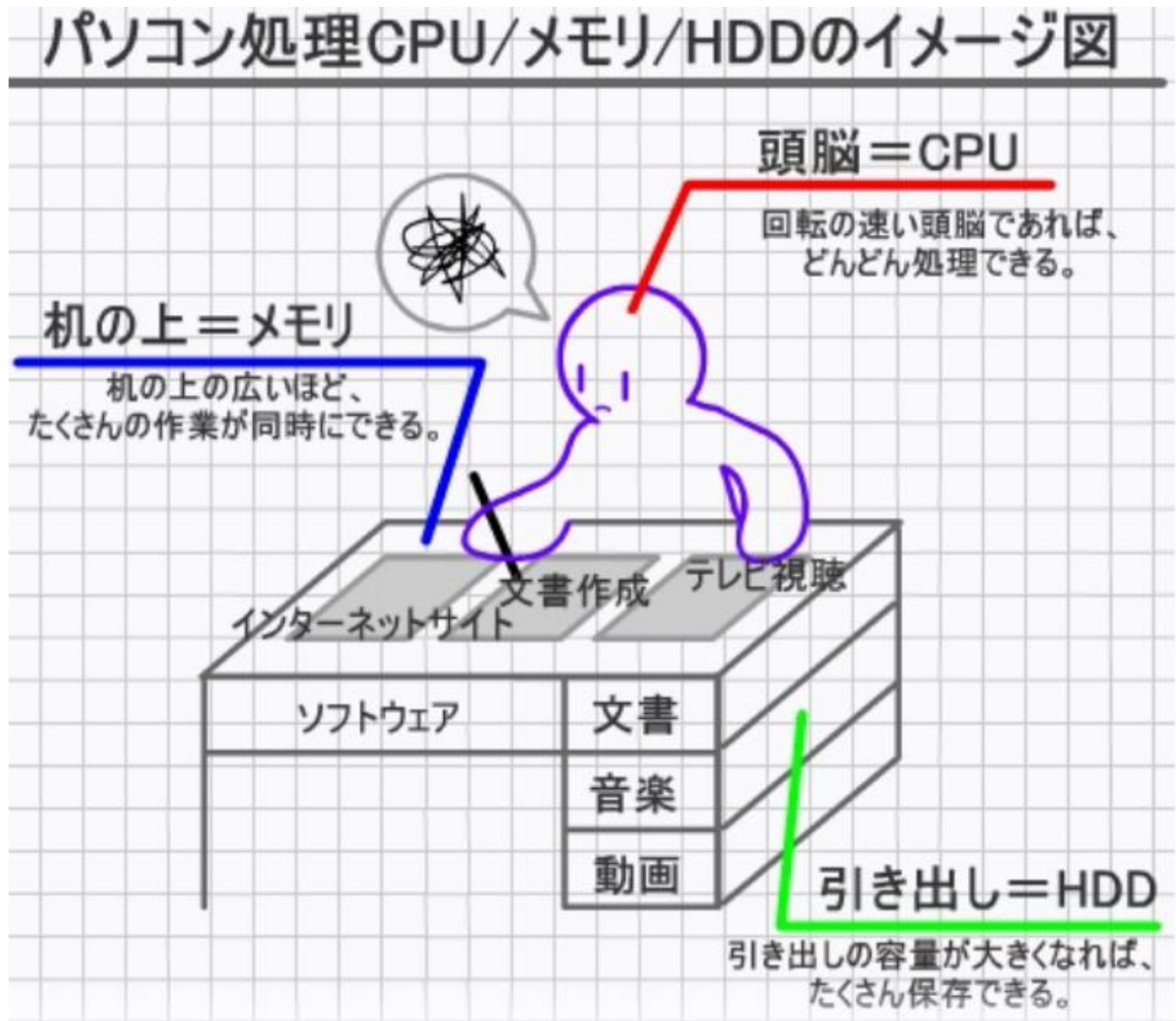


OSとは、「**オペレーティング・システム(Operating System)**」の略で、パソコンを操作したり、アプリを使ったりするための土台となる「**基本ソフトウェア**」です。

OSの仕事は「**端末全体の管理・制御**」であり、**ユーザーがパソコンやスマートフォンを簡単に操作できるように、重要な役割を果たしています。**

パソコンを「仕事部屋」とすると

OS を例えるなら「机で仕事をする人」



OS を搭載していないパソコンは電源を入れても何も表示されず、キーボードから文字を打ち込むこともできません。

つまり、ユーザーが当たり前のように使用しているパソコンやスマートフォンといった端末は、OS が入っていないければ、ただの空の箱と同じです。

パソコンやスマートフォンといった端末は、OS がインストールされて初めて使えるようになり、自由に操作することができます。

ここでいう操作の例は以下のようなものです。

- ・ キーボードやマウスからの入力を受け付けて画面に出力する
- ・ メモリの使用状況を管理する
- ・ CPU を効率的に使用できるように制御すること

ちなみに、ソフトウェアという特性上、1 台のパソコンに複数の OS を入れることもでき、パソコンの起動時に OS を選択することも可能です。

Unix システム

- **Unix** オペレーティングシステムは、大学や工学部のワークステーションとして、あるいは企業や組織のサーバとして使用されてきた長い歴史があります。
- 初期の OS である Unix は、GUI（図形画面）を持たず、すべての操作を**コマンドライン**から行っていました。
- Unix をベースとし、**UNIX 系 OS** と呼ばれる OS は、最新のものも含め、非常に多く存在します。その代表例が、オープンソースの Linux や Apple の macOS です。

GUI:画面にアイコンやボタンを表示して、それらをマウスなどのポインティングデバイスで操作

して命令を伝える

Linux システム

- **Linux** は、無料でオープンソースの UNIX 系オペレーティングシステムです。通常、Linux 自体はシステムではなく、Ubuntu、LinuxMint などの Linux カーネルをベースとしたオペレーティングシステムを総じて Linux と呼んでいます。
- Linux システムの主なメリットのは、その効率性とハードウェアの互換性です。そのため、**サーバ**などの大規模な計算機システムで幅広く利用されています。その上、無料で自由に安全なシステムであり、個人で使用するシステムの選択肢としても優れています。



Linux は Windows 、 macOS に次いで、世界で広く使われている OS です。

Linux の特徴は、**無償で配布**されていることと、**自由にカスタマイズ**ができること。

また、自分がカスタマイズしたものを再配布することもできます。

しかし、Linux を使いこなすには専門的な知識が必要で、OS の学習用で使う方も多いです

Windows システム

- **Microsoft Windows** は、Microsoft 社のグラフィカル・オペレーティングシステムで、世界で最も普及している PC 用の OS です。
- Windows は、macOS などの Unix 系 OS とは異なり、Microsoft 社自身が開発した MS-DOS をベースにしています。そのため、Windows のコマンドラインツール「cmd」は、macOS や Linux などとは全く異なる操作方法となりますが、近年、Microsoft は Unix に近いコマンドをサポートするツール「PowerShell」も導入しています。



OS の中でも世界中で広く知られて使われている Windows。

ビル・ゲイツによって開発され 1985 年に Microsoft 社が発表しました。

パソコンの OS ではシェア約 90%を占めています。

Windows 98、Windows2000、WindowsXP、WindowsVISTA、Windows7、Windows8、

Windows10 と次から次へと新しい OS を発信しています。

Mac システム

- **macOS** は、Apple 社が提供するグラフィカル・オペレーティングシステムです。こちらも主に PC 用のシステムとして使用されています。
- macOS は、主な競合相手である Windows と比較して、ビジュアル性能に重点を置き、より爽やかなユーザー体験 (UX) を提供しています。
- また、UNIX 系 OS である macOS は、コマンドライン操作の面で Linux と多くの類似点があります。



macOS は、Windows の次に世界で広く使われている OS で、1984 年に Apple 社が開発。

当時の OS は、上記画像のようにテキストベースの CUI が主流で、すべての操作をコマンドで手打ちしなければならず、一般ユーザーにとっては使いにくいものでした。

そのような時代に、macOS は初めて GUI を搭載した OS として登場し話題となりました。

つまり、今となっては主流の「マウスを使ってアイコンをクリックして操作ができる」ようにした革新的な OS なのです。

日本語キーボードレイアウト: Windows



日本語配列キーボードは、英語配列キーボードと違い「半角/全角」「無変換」「変換」「カタカナ/ひらがな」のキーがある。英語では必要のないキーのため、英語配列にはキーが存在しない。「半角/全角」などは日本語で多用されるため、日本では日本語配列キーボードが一般的！

他に、キーの並び方・キーの大きさなどの違いがある。英語配列は、「;/:」、「/’」、「[]」、「+」 「-」などが同じキーや隣りのキーにある。「スペースキー」は、「無変換」「変換」「カタカナ/ひらがな」がない分、大きい。「バックスペースキー」や「右 Shift キー」なども大きい。「Enter キー」は日本語配列は逆 L 字型で、英語配列は横長である。よく使うキーでは「@」が位置も違い、日本語配列はそのまま入力できるが、英語配列では Shift キーとの合わせ押しになるなどの違いもある。

日本語キーボードレイアウト: Mac



Mac の JIS 配列キーボードは、かなり US 配列に倣った形。なので、Windows に比べてそこまで差異がないことが分かります。違いとしては、**英数**キー・**かな**キーの有無、そして **Caps Lock** キーの位置があります。

Mac 専用ショートカットキー

| 名称 | 記号表記 | 対応する Windows のキー |
|-----------|------|------------------|
| Command | ⌘ | Ctrl |
| Shift | ⇧ | Shift |
| Option | ⌥ | Alt |
| Control | ⌞ | Ctrl |
| Tab | ⇥ | Tab |
| Caps Lock | ⇪ | Caps Lock |
| Delete | ⌫ | Backspace |
| Return | ↵ | Enter |

ファイル管理でよく使うショートカット

| 機能 | Windows | Mac |
|----------|--------------|-------|
| コピー | Ctrl + C | ⌘ + C |
| 貼り付け | Ctrl + V | ⌘ + V |
| 切り取る | Ctrl + X | ⌘ + X |
| 全選択 | Ctrl + A | ⌘ + A |
| 連続・単独選択 | Shift / Ctrl | ⇧ / ⌘ |
| 削除 | Delete | ⌘ + ⌫ |
| 操作を一つ戻す | Ctrl + Z | ⌘ + Z |
| 操作を一つ進める | Ctrl + Y | ⌘ + Y |
| ファイルを開く | Enter | ⌘ + ↓ |
| フォルダを開く | Enter | ⌘ + O |
| 親フォルダに戻る | Backspace | ⌘ + ↑ |

ドキュメント編集でよく使うショートカット

| 機能 | Windows | Mac |
|----------|--------------------|-------|
| 保存 | Ctrl + S | ⌘ + S |
| コピー | Ctrl + C | ⌘ + C |
| 貼り付け | Ctrl + V | ⌘ + V |
| 切り取る | Ctrl + X | ⌘ + X |
| 全選択 | Ctrl + A | ⌘ + A |
| 削除 | Delete / Backspace | ⌘ |
| 検索 | Ctrl + F | ⌘ + F |
| 操作を一つ戻す | Ctrl + Z | ⌘ + Z |
| 操作を一つ進める | Ctrl + Y | ⌘ + Y |

プログラミングでよく使うショートカット

| 機能 | Windows | Mac |
|------------|--------------------|---------------|
| 右にインデント | Tab | → |
| 左にインデント | Shift + Tab | ⇧ + → |
| コメントアウト・イン | Ctrl + / | ⌘ + / |
| キャレット移動 | Ctrl + ← / → | ⌘ + ← / → |
| 行をコピー * | Alt + Ctrl + ↑ / ↓ | ⌘ + ⌘ + ↑ / ↓ |
| 行を上下移動 * | Alt + ↑ / ↓ | ⌘ + ↑ / ↓ |
| リネーム * | Shift + Alt + R | ⌘ + ⌘ + R |
| コードの実行 * | Ctrl + F11 | ⌘ + F11 |
| 自動フォーマット * | Shift + Ctrl + F | ⇧ + ⌘ + F |

システムショートカット

| 機能 | Windows | Mac |
|-----------------|-------------------------------|------------------|
| アプリの切り替え | Alt + Tab | ⌘ + ⇧ |
| アプリを終了 | Alt + F4 | ⌘ + Q |
| 入力言語の切り替え | Alt + Shift / Ctrl + Shift | ^ + ⌘ + Space |
| タスクマネージャーを開く | Ctrl + Shift + Esc | ⌘ + ⇧ + Esc |
| コマンドライン 強制終了 | Ctrl + C | ^ + C / ⌘ + . |

コマンドラインとは

- **コマンドライン** [Command Line Interpreter, CLI] は、人間がキーボードでコマンドを入力し、コンピュータが文字で結果を出力し、人間とコンピュータの対話を実現するソフトウェアです。
- 画像形式のインタフェースによる直感的な操作に比べ、コマンドラインはコマンドを覚える必要があり、入門が難しいです。しかし、基本的なコマンドさえマスターしておけば、ファイルのバッチ処理やプログラミング関連の操作をより素早く、効率的に行うことができるようになります。
- 本日は、Windows の `cmd.exe` と macOS の `Terminal` を紹介します。



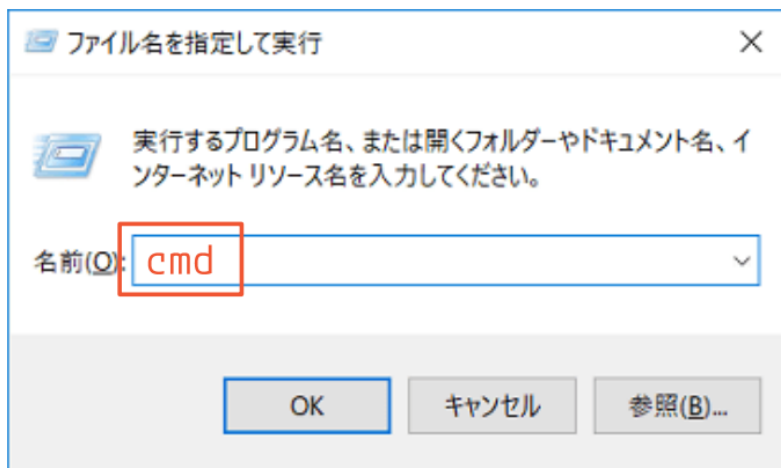
cmd.exe



Terminal

Windows CMD の開き方法

- Windows + R で「実行」を開きます。（またはメニューバーの検索ボックスを開きます。）
- 検索ボックスに「cmd」と入力し、Enter キーを押します。



Mac Terminal の開き方法

- 左下にある Launchpad を開き、



- 検索ボックスに terminal と入力し、開きます:



パス

- **パス** [Path] とは、フォルダシステム内のあらゆる **ファイル** [File] または **フォルダ** [Folder] の場所を表すものです。絶対パスと、相対パスの 2 種類があります。
- パスが表現する構造自体は **ディレクトリ** [Directory] と呼んでいます。その 2 つの概念は意味が近いので、よく混用されています。

コマンドラインインターフェースの概要

- Windows CMD (PowerShell) :

```
C:\Users\Brennan>_
```

ドライブ パス (フォルダ名) カーソル
レター

- macOS:

```
john@MacBook-Pro ~ %
```

ユーザー デバイス名 パス カーソル
ネーム

- 「cd」 コマンドをタイプし、コマンドラインのやりとりを観察しましょう。

Note !

日本語システムには
「\」記号がすべて
「¥」で表示される
ものがあります。

パス

- **パス**^[Path]とは、フォルダシステム内のあらゆる**ファイル**^[File]または**フォルダ**^[Folder]の場所を表すものです。**絶対パス**と、**相対パス**の 2 種類があります。
- パスが表現する構造自体は**ディレクトリ**^[Directory]と呼んでいます。その 2 つの概念は意味が近いので、よく混用されています。

絶対パスと相対パス

- **絶対パス** [Absolute Path]: システムの**ルート** [Root] ディレクトリ (Windows の場合は「C:\」「D:\」、Mac の場合は「~/」) から開始し、「\」または「/」で区切りして、すべてのフォルダ名を**外側から内側**に列挙します。
- **相対パス** [Relative Path]: **現在のフォルダからのパス**です。特殊な表記法として:
 - 「.」は、現在のフォルダを表現できます。
 - 「..」は、外側のフォルダ（親フォルダ）を表現できます。
- 質問: cd コマンドを使用して、**現在いるフォルダ外**のフォルダに移動するにはどうすればよいですか? **2 階層外**のフォルダに移動するにはどうすればよいですか?

よく使うコマンド (1)

| コマンド | CMD | Terminal |
|------------|--------------|-------------|
| ディスクの切替 | [ディスクレター]: | |
| フォルダに移動 | cd [パス] | cd [パス] |
| フォルダ情報を表示 | dir | ls |
| ファイルを開く | [ファイル名] | [ファイル名] |
| ファイルの内容を表示 | type [ファイル名] | cat [ファイル名] |
| コマンド一覧を表示 | man | help |
| コマンドの説明を表示 | man [コマンド] | help [コマンド] |

Note

Windows で、他のコマンドを実行する前に、**必ず正しいディスクに切り替えてください。**

よく使うコマンド（2）

| コマンド | CMD | Terminal |
|--------|------------------------|-------------------|
| ファイル作成 | echo > [ファイル名] | touch [ファイル名] |
| フォルダ作成 | md [フォルダ名] | mkdir [フォルダ名] |
| ファイル移動 | move [ファイル名] [目標パス] | mv [ファイル名] [目標パス] |

Note

名前にスペースがある場合は、「**"**」で囲む必要があります。

よく使うコマンド（3）

| コマンド | CMD | Terminal |
|----------|------------------------------|------------------------------|
| ファイルをコピー | copy [元ファイル名] [新しいファイル名] | cp [元ファイル名] [新しいファイル名] |
| フォルダをコピー | xcopy [元フォルダ名] [新しいフォルダ名] | cp -R [元フォルダ名] [新しいフォルダ名] |
| ファイルを削除 | del [ファイル名] | rm [ファイル名] |
| フォルダを削除 | rd [フォルダ名] | rmdir [フォルダ名] |

そもそも、パスって何？

パス（path）とは日本語に訳すると”道”や”経路”などといった意味です。
パスの先には目的のファイルが存在しており、そのファイルまでの道順（階層構造）を示したものがパスと呼ばれています。
分かりやすく言うと、目的のファイルまでの住所と言えることができるよ！

パスの種類

パスの種類は2種類で、絶対パスと相対パスがあるよ！！

絶対パス(Absolute path)

絶対パスとは・・・

絶対パスとは、フルパスとも言われており、最上位に位置するディレクトリ（ルートディレクトリ）から、対象のファイルまでの道順(階層構造)をすべて記述する方法になります。

https://ドメイン/フォルダ名/ファイル名のように指定し、ドメインから対象のファイルまでの経路を記述します。

この記述方法は、ユーザーがどのページにいても変わることなくファイルを表示することが可能です。

【ポイント】

- ・階層が深くなるごとに「/」で区切る。
例) http://ドメイン/フォルダ名/フォルダ名/ファイル名
- ・https://～から始まるパスは絶対パス
- ・最上位に位置するディレクトリから目的のファイルまでの道順を記述

相対パスとは…

相対パスとは**現在いるファイルから対象のファイルまでの道順(階層構造)**を記述する方法になります。

目的のファイルが同じ階層（カレントディレクトリ、カレントフォルダ）に存在する場合は**./ファイル名**

1つ上の階層に存在するファイルを指定する場合は

../フォルダ名/ファイル名

階層が1つ上に行く度に「../」は増えていき、../../フォルダ名/ファイル名となります。

同じ階層に存在するファイルを指定する場合は「./」は省略することが可能です。

【ポイント】

- 階層を1つさかのぼるごとに「../」を増やしていく
- 現在いるファイルから目的のファイルまでの道順を記述

【絶対パスのメリット】

- 全体から見た位置関係が明確になる
- 外部サイトへリンクを飛ばすとき（絶対パスのみ使用可能）
- リンク切れを起こしにくい

【相対パスのメリット】

ローカル環境でも作動する

絶対パスにくらべ短い記述で済むので、作業効率が上がる

SSL化やドメインを変更した際も、サイト内の構造が変わらないかぎり
は修正する必要がない