**IT**→「情報技術」のこと。コンピュータやネットワークを活用し社会に役立てる技術のこと。

**OS**→サーバやPC、スマホの頭脳。コンピュータを動かすためのソフトウェア。

主なOS

* **Windows**・・・Microsoft社が作ったOS。一般的なコンピュータにインストールされているOS。
* **Mac OS**・・・Apple社が作ったOS。
* **UNIX**・・・サーバのOS。現存する中でも古いOS。安定性が高く，ネットワーク機能に優れているため、長時間にわたって動かす必要のあるコンピュータなどに利用されている。
* **Linux**・・・サーバのOS。動作が早い。フリーソフトウエアとして公開されており、誰でも自由に開発、入手、使用などができる

**HTML**→webページを作成（構成）するための言語。

**CSS**→WEBページのレイアウトを指定する言語。

**JavaScript**→WEBページに動きを付ける言語。

**MySQL**→データベース管理システム。言語は「SQL」。MVCの「M」（model）の部分。

**PHP**→データベース（MySQL）とサイトと（Viewの部分）を繋ぐ言語。（MVCの「C」(controller)の部分。

**jQuery**→JavaScriptをより扱いやすくしたファイル（ライブラリ）。コードの記述を短縮できる。

**Apache**→Webサーバソフトウェア（Webサイトを提供するためのソフトウェア）の一つ。

**Bootstrap**→Webサイト構築用のWebフレームワーク。レスポンシブWebデザインに対応している。

**SE**→情報システム開発における上流工程（システムの設計や構築）を担当する職種。

**PG**→コンピュータシステムのプログラミングを行う職種。

**WEBデザイナー**→Webサイトのデザインを考える職種。

**CUI**→キーボードのみで操作するインターフェース。

**GUI**→マウスやタッチパットを使って操作するインターフェース。

**ファイル**→データのまとまり

**ディレクトリ**→フォルダのこと。

**ルートディレクトリ**→ディスクを開いて最初にフォルダがある場所、階層構造の一番上にある場所のこと

**カレントディレクトリ**→現在の（作業中の）ディレクトリのこと

**パス**→ファイルやディレクトリの場所を示すもの

**絶対パス**→ルートディレクトリから（ディスクの一番上から）の経路のこと。

**相対パス**→カレントディレクトリ（現在のディレクトリ）からの経路のこと。相対パスは絶対パスよりも短く表記することができるため、ホームページ作成によく使われる。

**HTTP**→WebサーバとWebクライアントの間でデータの送受信を行うために用いられるプロトコル（通信規約・通信上のルール）。HTML文書や画像などのデータをWebサーバとWebブラウザ間でやり取りするために使われる。HTTPでは、データを受信する側（クライアント）が要求をサーバに伝え、それに対してサーバが応答する。

**FTP**→ネットワークを経由してファイルを転送するための通信プロトコル。FTPサーバ、FTPクライアントの二種類のソフトウェアを用い、両者の間で接続を確立してファイルを送受信することができる。

**HTTPS**→Webのデータ転送に用いられるHTTPが、SSLやTLSで暗号化されている状態を表したもの。WebサーバとWebブラウザの間の通信が暗号化されていることを意味し、通信経路上での盗聴や第三者によるなりすましを防止する。

**SSL**→ネットワークでデータを暗号化して送受信するプロトコル。データを送受信する一対の機器間で通信を暗号化し、中継装置などネットワーク上の他の機器による成りすましやデータの盗み見、改竄などを防ぐことができる。

**SSH**→ネットワークを介して別のコンピュータにログインして遠隔操作するためのソフトウェアの一つ。主にUNIXコンピュータで利用される。通信途中の情報（通信経路）が暗号化されるため、インターネットなどを経由しても安全にアクセスすることができる。

**ポート**→パソコンと周辺機器をケーブルで接続するために使用する端子のこと。ケーブルやコネクタの差込口など。

**ポート番号**→ポートを識別するために割り振られた番号のこと。ポートがドアなら、ポート番号はドアの番号のようなもの。

主なポート番号

* HTTP TCP 80
* SSH TCP 22
* FTP (データ) TCP 20 (データのやり取り。実際にファイルデータを転送するポート)
* FTP (制御) TCP 21 (FTPサーバとのやり取りをする際の開始と終了の制御)

**ファイアウォール**→ネットワークの門番。コンピュータやネットワークと外部ネットワークの境界に設置され、内外の通信を中継・監視し、外部の攻撃から内部を保護するためのソフトウェアや機器、システムなどのこと。

**TCP**→インターネットにおいて標準的に利用されているプロトコル。IPと、HTTPなど上位プロトコルとの橋渡し係。送ったデータが相手に届いたか、その都度確認しながら通信するやり方。いちいち確認するため、信頼性は高いが転送速度が低いという特徴がありインターネットでよく利用される。

**IPアドレス**→ネットワークに接続されたコンピュータや通信機器の一台ごとに割り当てられた識別番号。インターネットをするときにコンピュータに割り当てられる住所のことで、通信においてデータの宛先の指定や送信元の特定などに用いられる。

IPアドレスの種類

* **グローバルIPアドレス**→全世界で通用するIPアドレスで、インターネット上で使用するアドレス。1台1台にグローバルIPアドレスが割り当てられている。
* **プライベートIPアドレス**→特定の組織内ネットワークのみで通用するIPアドレスでインターネットに接続しない。

**ドメイン**→インターネット上の「住所」。IPアドレスを文字列に置き換えたもの。

**DNS**→IPアドレスとドメインを紐づけをするシステム。必要に応じてIPアドレスをドメインに、ドメインをIPアドレスに変換してくれる通訳者。

**VPN**→、通信事業者の公衆回線を経由して構築された仮想的な組織内ネットワーク。また、そのようなネットワークを構築できる通信サービス。インターネット上に仮想の専用線を設け、安全なルートを確保した上で情報をやり取りすることにより、データの盗聴や改ざんなどから情報を守ることができる。

**ブリッジ**→ネットワークの中継機器。ネットワーク中のデータを見て、行き先を振り分けてくれる機器のこと。

**プロキシサーバ**→ホームページを見るときに使うWebブラウザの身代わりになってホームページにアクセスしてくれるコンピュータ（サーバ）また、そのような機能を持つサーバソフトウェアのこと。

**SMTP**→、電子メール（eメール）を送信するための通信手順（プロトコル）のこと。

**POP**→電子メール（eメール）を受信するための通信規約（プロトコル）のこと。直接見るIMAPに対してPOPは「メールのコピーをパソコンに持ってきて見る」方法。

**IMAP**→電子メール（eメール）を受信するための通信規約（プロトコル）のこと。POPと違いIMAPは「メールサーバにお邪魔して、箱に入っているメールを直接見る」方法。

**セッション**→コンピュータシステムやネットワーク通信において、接続してから、切断するまでの、一連の操作や通信のこと。

**クッキー**→Webサイトの提供者が、Webブラウザを通じて訪問者のコンピュータに一時的にデータを書き込んで保存させる仕組み。（買い物の履歴など）

**キャッシュ**→読み込んだデータのうち、直近に読み込んだものや使用頻度が高いものブラウザなどが、ウェブページの画像やデザイン情報を一時保管し次に同じページにアクセスしたときに素早く表示してくれる仕組み。

**チューニング**→コンピュータやソフトウェアなどの設定や構成を調整し、目標の状態に近づけたり、性能を最大限引き出したりする作業。具体的には、通信の状態が良くなるように電波状況を調整したり、たくさんのデータを処理するためにプログラムの作りを見直したりなど。

**スロークエリ**→データベースにおいて、時間のかかっているSQL(遅いSQL)のこと

**インデックス**→データベースでデータを検索する際、テーブルに格納されているデータを 高速に取り出す為の仕組み。最初にあらかじめ作成しておいたインデックスを検索することで、高速化を図る。

**フロー**→プログラムを構成する命令や手続き、処理の流れなどのこと。これを図にしたものがフローチャート図。

**フローチャート図**→処理の流れ（フロー）を、箱と線で表現したもの。コンピュータプログラムの設計やアルゴリズム（計算手順）の理解などのために、内部で行われる処理や演算の詳細な流れをフローチャートに表すことが多い。

**チケット管理ツール**→チームメンバーやコスト、スケジュールといったプロジェクトに関わる情報を管理するツール。代表的なのは、「Redmine」。

**タスク**→コンピュータが処理を行う際の作業単位のこと。

**Git**→ファイルの変更履歴を管理してくれるバージョン管理システムのひとつ。個々のファイルにいつ誰がどのような変更を行ったかを記録しており、必要に応じて特定の日時の版を参照したり、その状態に戻したりすることができる。

**Gitホスティングサービス**→Gitサーバーを貸し出し利用させるサービス。代表的なのは、「GitHub」。

**SourceTree**→Gitを簡単に使えるようにしたツール。無料で使える。

**SVN（Subversion）**→ファイルの変更履歴を管理してくれる、バージョン管理システムの一つ。複数の開発者が共同で一つのソフトウェアを開発する際などに、ソースコードやドキュメントを管理するのに用いられる。

**コンフリフト**→競合という意味。複数のプログラムが同じデータやファイルを同じタイミングで使おうとしてぶつかりあい、動作が不安定になる状態のこと。

**デグレ**→新しいバージョンのソフトウェアの品質が、以前より悪くなること。また、以前修正した不具合やバグが再発・復活すること。また、新しいファイルなどを古い内容で上書きしてしまい、更新内容が失われること。

**マージツール**→バージョン管理されているファイル群に対して行われた変更を統合する操作のこと。

**ロジック**→プログラムを処理するための内容や手順のこと。動作の流れを決めることは「ロジックを組む」と呼ばれ、このロジックに従ってプログラミングを進めていく。

**デバッグ**→テストなどによって発見された誤作動・不具合について、その原因やプログラム上での位置を探索・特定し、意図したとおり動作するように修正する作業のこと。

**開発手法**→業務システムの開発工程を構造化し、計画・制御するための枠組みのこと。

* **ウォーターフォール開発**→システム開発の手順の一つで、設計やプログラミングといった各工程を一つずつ順番に終わらせ、次の工程に進んでいく方式。最も古典的な開発モデルの一つで現代でも広く普及している。
* **アジャイル開発**→ソフトウェアを迅速に、また、状況の変化に対して柔軟に対応できるよう開発する手法の総称。設計→製造→テストのような一発勝負ではなく、作る→見てもらう→作り直す……のように「作りながら考える」というやり方のこと。

**アルゴリズム**→問題解決にいたる「やり方」や「考え方」。コンピュータにプログラムの形で与えて実行させることができるよう定式化された、処理手順の集合のこと。

**設計書**→設計した内容をまとめた資料。作る過程の説明を書いた資料。

* **外部設計**→依頼を受けて、全体像や概要をざっくりと考える工程のこと。利用者が必要とする必要な要件に基づいて、利用者から見てシステムがどのように振舞うべきかを決めていく。操作画面や操作方法（ユーザインターフェース）の構成や、帳票類の書式、データベースの構造などを決めることが多い。
* **内部設計**→「詳細設計」のこと。外部設計と実装の中間に位置し、実際のプログラムが作れるように、システムの詳細を考える工程。システムやソフトウェアをどのような構成にするか、それぞれの部分がどのような処理を行うべきか、それら部分間の連携・統合の方法などを決めることが多い。

**要件定義**→開発において、お客様の要望をまとめる工程。上流工程の一部で、実際の開発・実装作業を始める前に行う作業の一つ。実装すべき機能や満たすべき性能などのを明確にしていく。

* **要求仕様書**→要件定義の結果をまとめた資料。

**システム開発におけるテスト**→システムに不具合がないかを確認する。実装が完了し、リリースに向かって行うテスト。不具合の発生は主に、要件定義や設計書段階であり、検出は受け入れテストかリリース後の本番環境であることが多い。

* **単体テスト**→ソフトウェアやシステムのテスト手法の一つで、単一の部品（合体前の部品）を対象に行うテスト。それぞれの機能がきちんと動くかを確認する。
* **結合テスト**→複数のモジュール（部品）を組み合わせて行うテスト。個々のモジュールの単体テスト後に行う。主にモジュール間の接続点（インターフェース）がうまく機能するかを確認する場合と、結合した状態で外部から見て一体として正しく機能するかを確認する場合がある。
* **総合テスト**→システム全体を対象に行われるテスト。単体テスト、結合テストがすべて終わったあとに仕上げとして行うテスト。製品として完成したものを本番とほとんど同じ環境でテストする。システムが全体として要求された仕様のとおりに動作するか、性能は十分かなどを検証する。

**テスト仕様書→**要件定義書の通りに機能するかどうか、テストするポイントをまとめたもの。システムやソフトウェアの機能が明確になり、機能が正しく作動するのかどうか、誰がテストしても正しく検証することができる

テスト仕様書の項目

1. テストを実施した環境
2. 実施するテストの内容
3. テストを実施するためのシステムの操作手順
4. テストの実行結果
5. 個々のテスト項目を識別するための番号や記号
6. テストを実施した年月日
7. テストを実行した担当者
8. 障害報告票番号（発生した障害の詳細を開発グループに報告する際の識別番号）

**バリデーション**→入力場所、入力内容、書式設定、データの形式等が、仕様や文法などに照らして適切に記述・構築されているかどうかを確認すること。

**FM(フレームワーク)**→システム開発を楽に行えるように用意された、プログラムや雛形のこと。

**PHPのFW（フレームワーク）**

主なFW

* **Laravel**・・・最も注目されているPHPフレームワーク。処理速度はそれほど高くないが、コードが書きやすく、また見通しがいいことが特徴。
* **CakePHP**・・・簡単に動作する点が特徴。ルールがしっかりとしている分、CakePHP特有の書き方に慣れる必要あり。
* **Symfony**・・・スピードよりも機能を重視する場合や多人数で規模の大きな開発に向いているフレームワーク。アプリケーションを開発する上で開発と保守を効率化することを目的としており、さまざまなOS上で動作する。
* **Codeigniter**・・・軽量でコード量が少なく、比較的規模の小さなアプリケーションに向いているフレームワーク。

**Ajax**→JavaScriptを使ってWebページを表示した状態のまま、別のページや再読込などを伴わずにWebサーバー側と通信を行い、表示内容を変更する手法。画面全体を再表示すること無く、表示内容を新しいページの内容に書き換えることができる。

**API**→コンピュータプログラム（ソフトウェア）の機能や管理するデータなどを、外部の他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約のこと。ソフトウェアの一部をWEB上に公開することによって、誰でも外部から利用することができるようになり、それによって、自分のソフトウェアに他のソフトウェアの機能を埋め込むことができるようになるため、アプリケーション同士で連携することが可能になる。

**LAMP**→Linux＋ Apache ＋ MySQL（MariaDB） ＋ PHP（Perl、Python）で頭文字を繋いだもの、構成される環境のこと。

**クラウドサーバー**→クラウド上にある仮想サーバ。オンライン環境とアカウントがあれば、どこにいてもサーバにアクセスでき、いつでも観賞したりダウンロードしたりできる。

**レンタルサーバー**→サーバーをレンタルして使用するサーバー。1台のサーバーを複数サイトで共有する。

**VPS**→物理的には1台のサーバを複数の人で使うが、利用者ごとのスペースは独立している形態になっているレンタルサーバのこと。

**AWS**→アマゾンウェブサービスのこと。Amazonが事業者向けに提供しているクラウドサービス。

**プラットフォーム**→動かすために必要な、土台となる環境。具体的には、ある機器やソフトウェアを動作させるのに必要な、基盤となる装置やソフトウェア、サービス、あるいはそれらの組み合わせ（動作環境）のこと。

**カバレッジ**→カバーしている範囲のこと。テスト全体で、これくらいの確認をする。

**エビデンス**→システム開発などで、最終検証段階においてシステムが発注どおりに動作していることを示す証拠となる書類やデータなどのこと。

・テストに使ったデータ

・プログラムを動かした結果の画面（ハードコピー）

・プログラムを動かしたときの記録（ログ）

など。

**バッチ処理**→複数のプログラムからなる作業において、あらかじめ一連の手順を登録しておき、まとめて連続的に実行する方式。または、一定期間や一定量ごとにデータをまとめて一括して処理する方式のこと。

**デプロイ**→配置する、配備する、展開するなどの意味を持つ英単語。作ったプログラムを運用環境に展開や、設定をしたりして、使える状態（プログラムを動かせる状態）にすること。

**データベース接続クライアント**→データベース管理システム上のデータを容易に扱う為のソフトウェア。代表例として、「MySQL」。

**DBのJOIN**→データベースの結合。複数のテーブルのデータをまとめて1つのデータにすること。「a」のテーブルと「b」のテーブルのデータを1つにして、「c」のテーブルのデータにする。

**SEO**→サーチエンジンの検索結果のページの表示順の上位に自らのWebサイトが表示されるように工夫すること。また、そのための技術やサービス。