情報技術基礎

コンピュータ
:自動計算機

ハードウェア:
デジタル回路なども含めたコンピュータの物理的部分。入力装置、出力装置、処理装置、記憶装置、駆動装置などがある。

CPU:
(Central Processing Unit)コンピュータにおいて中心的な処理装置として働く電子回路。プログラムによって様々な数値計算などを行う

プロセッサ:
処理装置の総称。コンピュータ中で、プログラムに記述された命令セットを実行するためのハードウェアであり、演算装置、命令や情報を格納するレジスタ、周辺回路などから構成される。

レジスタ:
コンピュータのプロセッサなどが内蔵する記憶回路で、制御装置や演算装置や実行ユニットに直結した操作に要する。速度が最速だが少量。

キーボード:
コンピュータへの入力装置の一つ。指でキーを押すことでコンピュータへ文字信号などを送信するもの。

モニター:
ディスプレイ。コンピュータから出力される静止画または動画の映像信号を表示する機器である。

補助記憶装置:
データを保持する電子媒体と読み書きを行う駆動装置(=ドライブ)で構成される。ストレージともいう。

ハードディスク:
(Hard disk drive、HDD)情報を記録し読み出す装置。

SSD(Flash Solid State Drive):
RAMディスクと同様に半導体メモリであるフラッシュメモリ使用した単独の記憶装置。PC用やサーバー機での使用が進んでいる。

メモリ:
主記憶装置。プロセッサが直接アクセスすることのできる記憶装置。読み書き自由なRAMと、読み取り専用のROMがある。

ソフトウェア:
コンピューターシステム上で何らかの処理を行うプログラムや手続き。

システムソフトウェア:
コンピュータのハードウェアを管理・制御するなど、コンピュータの稼動や使用そのものをサポートするためのソフトウェア全般

オペレーティングシステム(OS):
コンピュータにおいて、ハードウェアを抽象化したインターフェースをアプリケーションソフトウェアに提供するソフトウェア

ミドルウェア:
OSとアプリケーションソフトウェアの中間に入るソフトウェア

言語処理系:
コンパイラやインタプリタ

コンパイラ:
プログラムのソースコードを、機械語、ないしバイトコードなどの中間言語によるオブジェクトコードに翻訳(変換)するプログラム。

インタプリタ:
ソースコードないし中間表現にある命令列を逐次解釈しながら実行するプログラム。

ファームウェア:
電子機器に組み込まれたコンピュータシステム（ハードウェア）を制御するためのソフトウェアで、ソフトウェアをROM等の集積回路などに書き込んだ状態で、機器に組み込んだもの

アプリケーションソフトウェア:
コンピュータの利用者がコンピュータ上で実行したい作業を実施する機能を直接的に有するソフトウェア。ワープロや表計算など。

ファイル:
主に補助記憶装置に格納されたデータを指すが、デバイスやプロセス、カーネル内の情報といったものもファイルとして提供するファイルシステムもある。

ファイルフォーマット(ファイル形式):
ファイルの保存形式。ファイルフォーマットの識別として拡張子をつける。しかし、拡張子は簡単に編集できるため信頼性は極めて低い。そのため、ファイルの先頭にマジックナンバーと呼ばれるより確実にファイル形式を識別するための情報を付加する。

ファイルシステム:
コンピュータのリソースを操作するためにOSが持つ機能の一つ。正確に定義すれば、ファイルシステムは抽象データ型の集まりであり、ストレージ、階層構造、データの操作/アクセス/検索のために実装されたものである。

ディレクトリ(フォルダ):
記憶装置で、ファイルを分類・整理するための保管場所。

絶対パス:
記憶装置は、ルートディレクトリを頂点とする木構造になっており、これに沿って頂点から目的のファイルやフォルダまでのすべての道筋を記述したもの

パス(PATH):
外部記憶装置内でファイルやフォルダの所在を示す文字列。ファイルやフォルダのコンピュータ内での住所にあたる。

相対パス:
起点を現在位置として、目的のファイルやフォルダまでの道筋を記述したもの

ネットワーク:
ネットワーク上で、各ノードを識別するために設定されているLANカードなどのネットワーク機器のハードウェアに一意に割り当てられる物理アドレス

MACアドレス:

ICANN(アイキャン):
インターネットのIPアドレスやドメイン名などの各種資源を全世界的に調整・管理することを目的として、1998年に設立された民間の非営利法人である。

ドメイン:
IPネットワークにおいて個々のコンピュータを識別する名称の一部。インターネット上においてはICANNによる一元管理となっており、世界中で絶対に重複しないようになっている。通常、IPアドレスとセットでコンピュータネットワーク上に登録される。

DNS(Domain Name System):
ドメイン名を含むホスト名とIPアドレスと対応を管理し、変換する。

ホスト名:
コンピュータ、ルータなどネットワークに接続された機器(ホスト)に付けられた名前

IPアドレス:
IPにおいてパケットを送受信する機器を判別するための番号

IP技術:
インターネットプロトコルの略

インターフェース:
界面や接触面、中間面などといった意味を持ち、転じてコンピュータと周辺機器の接続部分を表すようになった。ものごとの境界となる部分と、その境界での処理方式であるプロトコルを指す。

API:
ソフトウェアコンポーネントが互いにやりとりするのに使用するインタフェースの仕様

コンポーネント:
ソフトウェアシステムの様々な機能を関心の分離によって分割したものである。システムを独立した結合の弱い再利用可能なコンポーネント群で構成する設計技法。

SDK:
特定のプログラミング言語用のインターフェイスのためのファイル群という形式のAPIや、何らかの組み込みシステムと通信するためのハードウェアなど

関心の分離:
ソフトウェア工学において、プログラムを機能面において可能な限り重複がない、複数の機構に明確に分割すること。関心とは、プログラムのある機能や振る舞い、目的のことである。

デザインパターン:
設計パターン。過去のソフトウェア設計者が発見し編み出した設計ノウハウを蓄積し、名前をつけ、再利用しやすいように特定の規約に従ってカタログ化したもの

フレームワーク:
プログラミングにおいて、一般的な機能をもつ共通コードをユーザーが選択的に上書きしたり特化させたりすることで、ある特定の機能をもたせようとする抽象概念のこと

ライブラリ:
汎用性の高い複数のプログラムを、再利用可能な形でひとまとまりにしたもの。他のプログラムに何らかの機能を提供するコードの集まり。

組み込みライブラリ
数値(Number)、文字列(String)
配列(Array)、辞書(Dictonary/Hash)

標準添付ライブラリ:
デザインパターン、ファイルフォーマット、数学など開発で便利なもの

インターネット
:IP技術を利用した相互接続によるコンピュータネットワーク

プログラム
:コンピュータに対する命令処理を記述したテキスト

プログラミングパラダイム:
プログラマにプログラムの見方を与えるもの。例えば、オブジェクト指向プログラミングにおいて、プログラムとはオブジェクトをつくりそれを管理するものである。関数型プログラミングにおいては、状態を持たない関数の評価の連続である。

手続き型プログラミング:
実行すべき一連の計算ステップを持つもの。手続きはプログラム実行中の任意の時点で呼び出すことができ、他の手続きからの呼び出しも、自分自身からの呼び出しもできる。

関数型プログラミング:
引数に関数を作用させて計算をおこなう。「関数」とは「数学」の関数である。

オブジェクト指向プログラミング:
相互にメッセージを送りあうオブジェクトの集まりとしてプログラムを構成する技法

アルゴリズム:
問題を解くための手順を定式化した形で表現したもの

シンタックス:
語が文を構成する仕組み。構文規則

式:
言語によって定められた優先順位や結びつきの規定に則って評価される値、変数、演算子、関数の組み合わせ

文:
複数の処理を一つにまとめたもの。手続き(プロシージャ)

分岐、反復:
分岐はif, while。反復はfor, eachなどが使われる

スレッド:
CPU利用の単位。プロセスに比べて、プログラムを実行するときのコンテキスト情報が最小で済むので切り替えが早くなる。スレッドは、thread of execution(実行の脈絡)の略語。

プロセス:
プログラムのコードおよび全ての変数やその他の状態を含む。OSによっては、プロセスが複数のスレッドで構成される場合があり、命令を同時並行して実行する

値と型と名前:
名前が値を参照している。値にはデータ型がある。

演算子:
各種の演算をあらわす記号のこと。

変数:
データを一定期間記憶し必要なときに利用できるようにするために、データに固有の名前を与えること

ポインタ
あるオブジェクトがなんらかの論理的位置情報でアクセスできるとき、それを参照するもの

アドレス:
読み出すメモリ上の位置の一意な識別子(論理的位置情報)。

サブルーチン(関数):
サブルーチンが返す値は「戻り値」と呼ばれる。「引数」をとり、値を返すことから「関数」と呼ばれる。オブジェクト指向ではメソッドとして記述。

アクセサ:
メソッドの中で特にオブジェクトからフィールドの値など単一の値を出し入れするメソッドのこと

ゲッタ(getter):
値を取得するアクセサメソッド

セッタ(setter):
値を代入するアクセサメソッド

プロパティ:
フィールドとメソッドの中間的な存在

変数の種類
ローカル変数、インスタンス変数、private、protected、public(グローバル変数)、クラス変数(静的変数)などがある

スコープ:
有効範囲。ある変数や関数が特定の名前で参照される範囲(アクセス範囲)

名前空間:
冗長な命名規則を用いなくても名前の衝突が起こらないようにし、しかもそれを容易に記述できるようにするためだけの概念

生存期間:
いつ使えるか。

データ構造②:
コレクション、配列(Array)、辞書(Hash)などをプログラミングでは使う

オブジェクト:
プログラム上の手続きの対象を抽象化する概念。クラス、インスタンス、インヘリタンス(継承)、インターフェースなどで実装

結合度:
オブジェクト間の依存度のこと。オブジェクト間の依存性が低いことを疎結合と言われていて、あるクラスの実装の変更が、他のクラスに影響を及ぼさないようにすることで、疎結合にして、コードの拡張性と保守性を高めることが目的である。

凝集度:
クラスの機能と属性の関連性の強さ。関連する機能(メソッド)や属性(変数)をあちこちに分散させずに、適切なクラスに分割することで、凝集度が高くなる。凝集度を高くする目的は、システムの保守性、拡張性を向上させることである。

データ構造①:
計算機科学においてデータの集まりをコンピュータの中で効率的に扱うため、一定の形式に系統立てて格納するときの形式

データ:
コンピューターの処理の対象となる数値文字記号

情報:
意味のあるデータの複合体のこと

通信:
情報を伝達すること。

パケット通信:
端末からのデータをPADでパケットに変換して伝送し、交換設備の記憶装置に蓄積し、中継伝送路の空いている時間に送り出し、受信側の交換機の記憶装置に蓄積された後送出され、PADで元のデータに変換され相手先の端末に届く通信方式である。

パケット:
プロトコルが扱うデータの単位。TCP/IPで利用。

PAD:
(Packet Assembly Disassembl)
パケット組み立て分解装置

規約:
人々の協議によって決めた規則。

PDU:
プロトコル・データ・ユニット

プロトコル:
通信規約。データ通信を行うために、あらかじめ定めておく規約。信号送信の手順、データの表現法などを定める。

SSH:
暗号や認証の技術を利用して、安全にリモートコンピュータと通信するためのプロトコル

FTP:
ネットワークでファイルの転送を行うための通信プロトコル

ウェブ(World Wide Web、WWW):
インターネット上で提供されるハイパーテキストシステム。W3Cあるいは単にWebとも呼ばれる。インターネットの応用技術のひとつである。

ISP(プロバイダ):
インターネットへの接続を提供する組織のことである(相互接続する)。日本では電気通信事業者の一つとして位置づけられている。

URI:
一定の書式によってリソース(資源)を指し示す識別子。URL(場所)とURN(名前)の2種類がある。

URN:
リソースの「名前」を識別する。

URL:
リソースの「場所」を識別する。インターネット上のデータ(ウェブページなど)の場所(アドレス)を特定するための文字列

HTTP(HyperText Transfer Protocol):
ハイパーテキスト転送プロトコル。WebブラウザとWebサーバの間でHTMLなどのコンテンツの送受信に用いられる通信プロトコル

ハイパーテキスト:
複数の文書(テキスト)を相互に関連付け(リンク)、結び付ける仕組み

ハイパーリンク:
ハイパーテキストにおいて、複数の文書を結び付ける役割を担う参照

メールプロトコル:
SMTP, IMAP, POP3など

SMTP:
送信者からメールサーバへの送信に使われるプロトコル

IMAP:
受信者はメールをダウンロードするが、メールはメールサーバ上に保存したままで管理されるプロトコル

POP3:
受信者がメールサーバに保存されたメールをダウンロードし、ダウンロードがすんだメールはサーバから削除するプロトコル。

Webサーバー:
HTTPに則りウェブブラウザに対して、HTMLやオブジェクト（画像など）の表示を提供するサービスプログラム及び、そのサービスが動作するサーバコンピュータ。Apache, Nginxなどがある。

Webブラウザ:
ユーザーの選択したウェブリソースをサーバーに要求してブラウザウィンドウに表示することにより、ユーザーに提示する。通常、リソースは HTMLですが、PDFや画像など、他のタイプのリソースもある。リソースの場所は URI（Uniform Resource Identifier）を使用してユーザーが指定します。

HTML(Hyper Text Markup Language):
ウェブ上の文書を記述するためのマークアップ言語。ほかのドキュメントへのハイパーリンクを設定できるハイパーテキストなどの高度な表現力を持つ。

マークアップ言語:
コンピュータ言語の一種で、文章の構造(段落など)や見栄え(フォントサイズなど)に関する指定を文章とともにテキストファイルに記述するための言語である。文章に対するそれらの指定を「マークアップ」と呼び、マークアップを記述するための文字列を「タグ」と呼ぶ。

クライアント・サーバー・モデル:
コンピュータシステムにおける分散処理形態の一つ。サーバーはクライアントからの処理要求を受けて実際の処理を行い、その結果をクライアントに返すというモデル。

JavaScript:
プログラミング言語。ブラウザにはJavaScriptインタプリタがひとつの機能として実装されている

JavaScriptフレームワーク:
Angular, Ember, Backboneなど

DOM:
HTMLとJavaScriptなどの外部世界とのインターフェースをオブジェクトで表現したもの。DOMとマークアップはほぼ１対１の関係

CSS:
HTMLの要素をどのように修飾（表示）するかを指示する、W3Cによる仕様の一つ。文書の構造と体裁を分離させるという理念を実現する為に提唱されたスタイルシートの、具体的な仕様の一つ。

CSSフレームワーク:
Twitter Bootstrapなど

スタイルシート:
構造化文書などにおける表示形式を制御するしくみ。見栄えと構造を分離するという目的で提唱された

JSON:
軽量なデータ記述言語の1つ

Ajax:
XMLHttpRequest(HTTP通信を行うためのJS組み込み関数)で通信し、ブラウザの一部を動的に変更する(=非同期通信する)技術的アプローチ方法

データ記述言語:
コンピュータにおいて扱うデータを記述するための人工言語。マークアップ言語もこれに含まれる。

モバイル:
スマートフォンもクライアントである。WEBブラウザとは異なるがアプリごとでWebサーバーと通信する。

iOS:
アップル開発のOS

Android:
グーグル開発のOS。オープン化している。

セキュリティ:
技術仕様や実装方針はセキュリティに大きく依存している

認証:
正当性を検証する作業。例えば、ユーザ名とパスワードの組み合わせを使って、コンピュータを利用しようとしている人にその権利があるか確認すること。

Cookie:
ブラウザとHTTPヘッダの仕組み。データ保存領域。認証など状態管理で使用。

Session:
一連の通信。状態を保持するための実装方法。トークンをCookieに保存・通信し、サーバーサイドのセッションテーブルに保存されているものとマッピングする。

Token:
それ以上細かい単位に分解できない文字列の並び

OAuth:
あらかじめ信頼関係を構築したサービス間で、ユーザの同意のもとに、セキュアにユーザの権限を受け渡しする

SSL:
インターネットなどのTCP/IP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

公開鍵暗号方式:
対になる2つの鍵(公開鍵と秘密鍵)を使ってデータの暗号化・復号を行う暗号方式。

攻撃手法:
バグをついたものなどたくさんあるが、必要最低限のマナーとして実装上注意が必要。

セッションハイジャック:
ネットワーク上で一対の機器間で交わされる一連の通信を途中で乗っ取り、片方になりすましてもう一方から不正にデータを詐取したり操作を行なう攻撃。

XSS:
ウェブページのセキュリティ上の不備を意図的に利用して、サイト間を横断して悪意のあるスクリプトを注入させること

CSRF:
Webサイトにスクリプトやリダイレクトを仕込むことで、閲覧者に意図せず別のWebサイト上で何らかの操作(掲示板への書き込みなど)を行わせる攻撃手法

SQLインジェクション:
データベースと連動したWebサイトで、データベースへの問い合わせや操作を行うプログラムにパラメータとしてSQL文の断片を与えることにより、データベースを改ざんしたり不正に情報を入手する攻撃

ORM:
オーアールマッパー。オブジェクト指向のデータとデータベースシステムのデータの変換を行う。

SQL:
RDBMSにおいてデータの操作や定義を行うためのデータベース言語。

データベース:
特定のテーマに沿ったデータを集めて管理し、容易に検索・抽出などの再利用をできるようにしたもの。 OSが提供するファイルシステム上に直接構築されるものや、データベース管理システムを用いて構築されるものを含む。

キーバリュー型DBMS:
キーに対してバリュー(値)という単純な構造。Redisなど。

ドキュメント指向DBMS:
スキーマレスでデータ構造が柔軟なもの。MongoDBなど。

リレーションDBMS:
リレーションモデルに基づいた、データを表の形式で永続化し、かつデータ間の関連もまた表の形式で永続化できるDBMS。一般に関係(リレーション)は、表(テーブル)と呼ばれることが多い。