

プログラミング言語目録

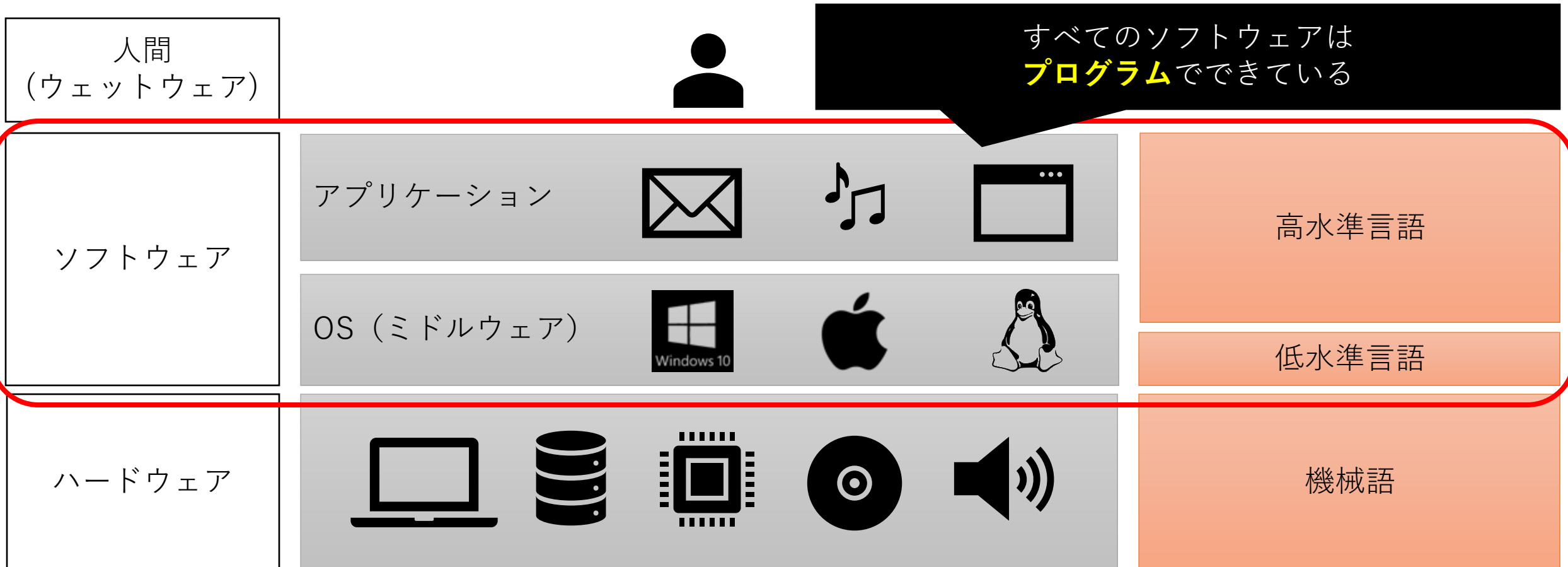
2025/05/13

青森大学 ソフトウェア情報学部

角田 均

プログラミングって何？

- プログラム = ソフトウェアを作ること



コンピュータとプログラム

- ノイマン型コンピュータ
 - 命令とデータ（=プログラム）をメモリに記憶させる
 - メモリにはアドレス（番地）がある
 - メモリ上の処理を順番に取り出し、CPUで実行する

メモリ1番地の命令

メモリ2番地の命令

メモリ3番地の命令

メモリ4番地の命令

```
1 print("おはよう")
2 print("こんにちは")
3 print("こんばんは")
4 print("おやすみ")
```

おはよう
こんにちは
こんばんは
おやすみ

上から順に、一つずつ実行される



ジョン・フォン・ノイマン
(1903-1957)

プログラムが動くまで

高水準言語
(C言語)

```
#include <stdio.h>

int main(void) {
    printf("Hello Word!\n");

    return 0;
}
```

```
.section      __TEXT,__text,regular,pure_instructions
.build_version macos, 10, 15      sdk_version 10, 15, 6
.globl _main                      ## -- Begin function main
.p2align     4, 0x90

_main:
.cfi_startproc
## %bb.0:
    pushq   %rbp
.cfi_def_cfa_offset 16
.cfi_offset %rbp, -16
    movq    %rsp, %rbp
.cfi_def_cfa_register %rbp
    subq    $16, %rsp
    movl    $0, -4(%rbp)
    leaq    L_.str(%rip), %rdi
    movb    $0, %al
    callq   _printf
    xorl    %ecx, %ecx
    movl    %eax, -8(%rbp)        ## 4-byte Spill
    movl    %ecx, %eax
    addq    $16, %rsp
    popq    %rbp
    retq
.cfi_endproc

L_.str:
.asciz "Hello Word!\n"

.subsections_via_symbols
```

低水準言語
(アセンブリ言語)

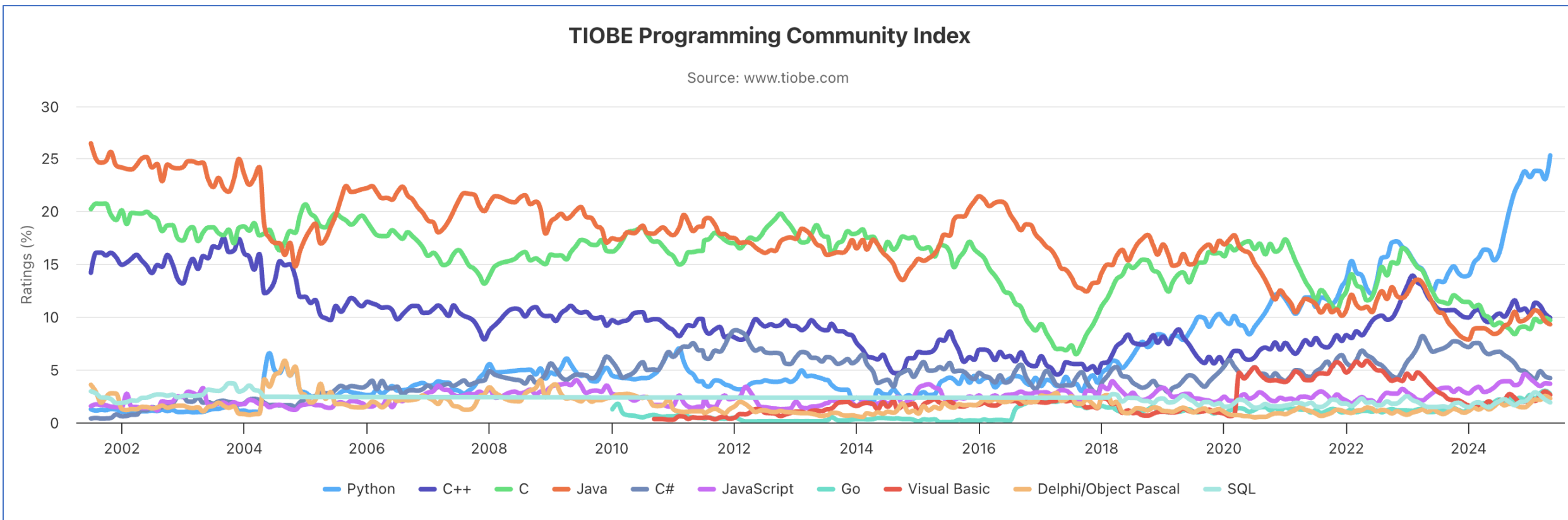
機械語 = 2進数
(デジタル回路のon/off)

```
00000000 cf fa ed fe 07 00 00 01 03 00 00 00 01 00 00 00
00000010 04 00 00 00 08 02 00 00 00 20 00 00 00 00 00 00
00000020 19 00 00 00 88 01 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040 98 00 00 00 00 00 00 00 00 28 00 00 00 00 00 00
00000050 98 00 00 00 00 00 00 00 00 07 00 00 00 00 00 00
00000060 04 00 00 00 00 00 00 00 00 5f 50 00 00 00 00 00
00000070 00 00 00 00 00 00 00 00 00 5f 50 00 00 00 00 00
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090 2a 00 00 00 00 00 00 00 00 28 02 00 00 00 00 00
000000a0 c0 02 00 00 02 00 00 00 00 04 00 80 00 00 00 00
000000b0 00 00 00 00 00 00 00 00 00 5f 5f 63 73 74 72 69 6e
000000c0 67 00 00 00 00 00 00 00 00 5f 5f 54 45 58 54 00 00
000000d0 00 00 00 00 00 00 00 00 00 2a 00 00 00 00 00 00 00
000000e0 0e 00 00 00 00 00 00 00 00 52 02 00 00 00 00 00 00
000000f0 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00
00000100 00 00 00 00 00 00 00 00 00 5f 5f 63 6f 6d 70 61 63
00000110 74 5f 75 6e 77 69 6e 64 5f 5f 4c 44 00 00 00 00 00
00000120 00 00 00 00 00 00 00 00 00 38 00 00 00 00 00 00 00 00
00000130 20 00 00 00 00 00 00 00 00 60 02 00 00 03 00 00 00 00
00000140 d0 02 00 00 01 00 00 00 00 00 00 00 02 00 00 00 00 00
00000150 00 00 00 00 00 00 00 00 00 5f 5f 65 68 5f 66 72 00 00
00000160 6d 65 00 00 00 00 00 00 00 5f 5f 54 45 58 54 00 00 00
00000170 00 00 00 00 00 00 00 00 00 58 00 00 00 00 00 00 00 00
00000180 40 00 00 00 00 00 00 00 00 80 02 00 00 03 00 00 00 00
00000190 00 00 00 00 00 00 00 00 00 0b 00 00 68 00 00 00 00 00
000001a0 00 00 00 00 00 00 00 00 00 32 00 00 00 18 00 00 00 00
000001b0 01 00 00 00 00 0f 0a 00 06 0f 0a 00 00 00 00 00 00 00
000001c0 02 00 00 00 18 00 00 00 d8 02 00 00 02 00 00 00 00 00
000001d0 f8 02 00 00 10 00 00 00 0b 00 00 00 50 00 00 00 00 00
000001e0 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00
000001f0 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000220 00 00 00 00 00 00 00 00 00 55 48 89 e5 48 83 ec 10 00 00
00000230 c7 45 fc 00 00 00 00 48 8d 3d 14 00 00 00 b0 00 00 00 00
00000240 e8 00 00 00 00 31 c9 89 45 f8 89 c8 48 83 c4 10 00 00 00 00
00000250 5d c3 48 65 6c 6c 6f 20 57 6f 72 64 6c 21 0a 00 00 00 00
00000260 00 00 00 00 00 00 00 00 00 2a 00 00 00 00 00 00 00 00 01
00000270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000280 14 00 00 00 00 00 00 00 01 7a 52 00 01 78 10 01 00 00 00
00000290 10 0c 07 08 90 01 00 00 24 00 00 00 1c 00 00 00 00 00 00
000002a0 88 ff ff ff ff ff ff 2a 00 00 00 00 00 00 00 00 00 00 00
000002b0 00 41 0e 10 86 02 43 0d 06 00 00 00 00 00 00 00 00 00 00
000002c0 19 00 00 00 01 00 00 2d 12 00 00 00 02 00 00 15 00 00 00
000002d0 00 00 00 01 00 00 06 01 00 00 00 0f 01 00 00 00 00 00 00
000002e0 00 00 00 00 00 00 00 07 00 00 00 01 00 00 00 00 00 00 00
000002f0 00 00 00 00 00 00 00 5f 6d 61 69 6e 00 5f 00 00 00 00 00
00000300 70 72 69 6e 74 66 00 00 00 00 00 00 00 00 00 00 00 00 00
00000308
```

Hello Word!

実行結果

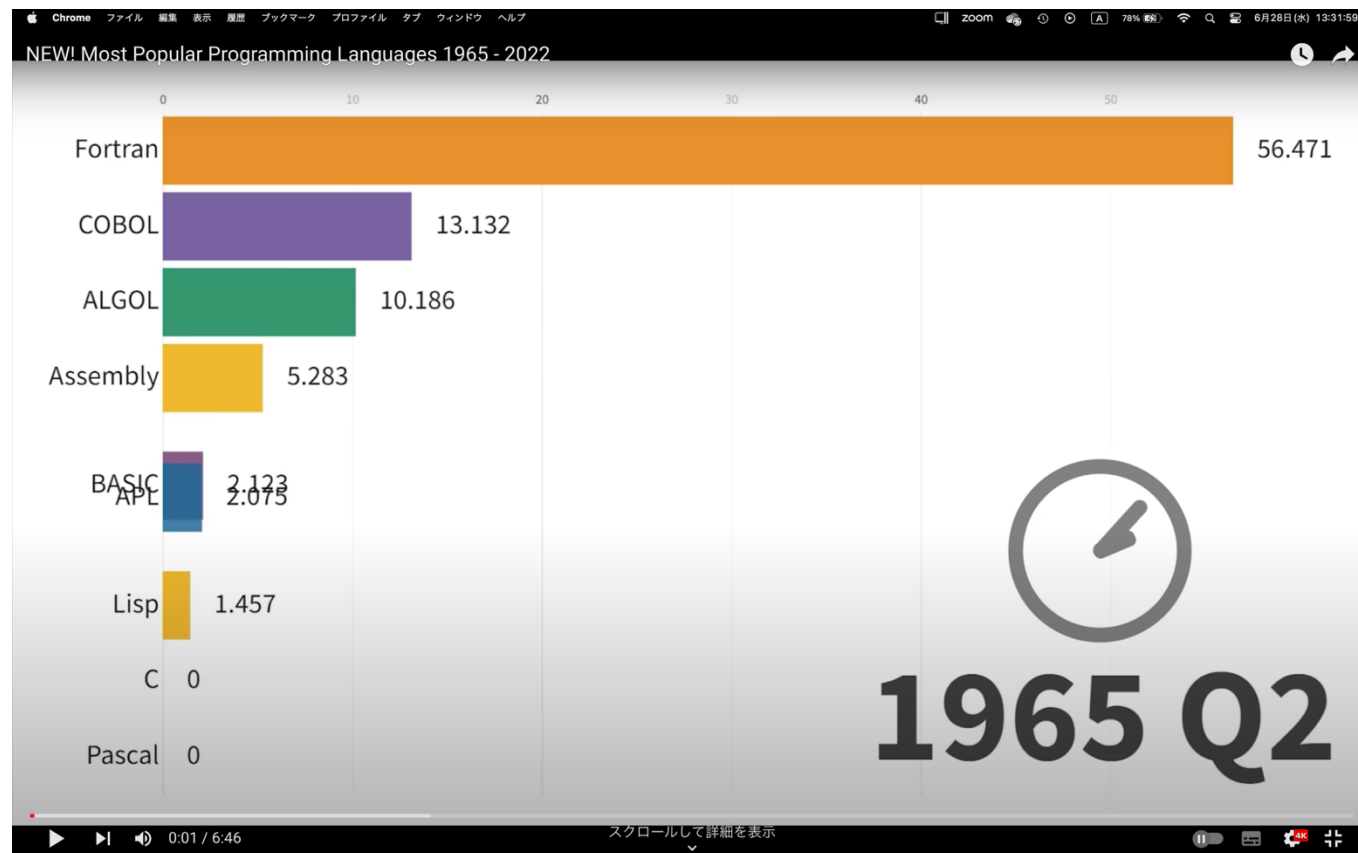
プログラミング言語の人気ランキング



Googleでの検索数、SNSでの発言数、学習コースの数などからレーティング

プログラミング言語の覇権抗争(半世紀)

- Most Popular Programming Languages 1965 - 2022

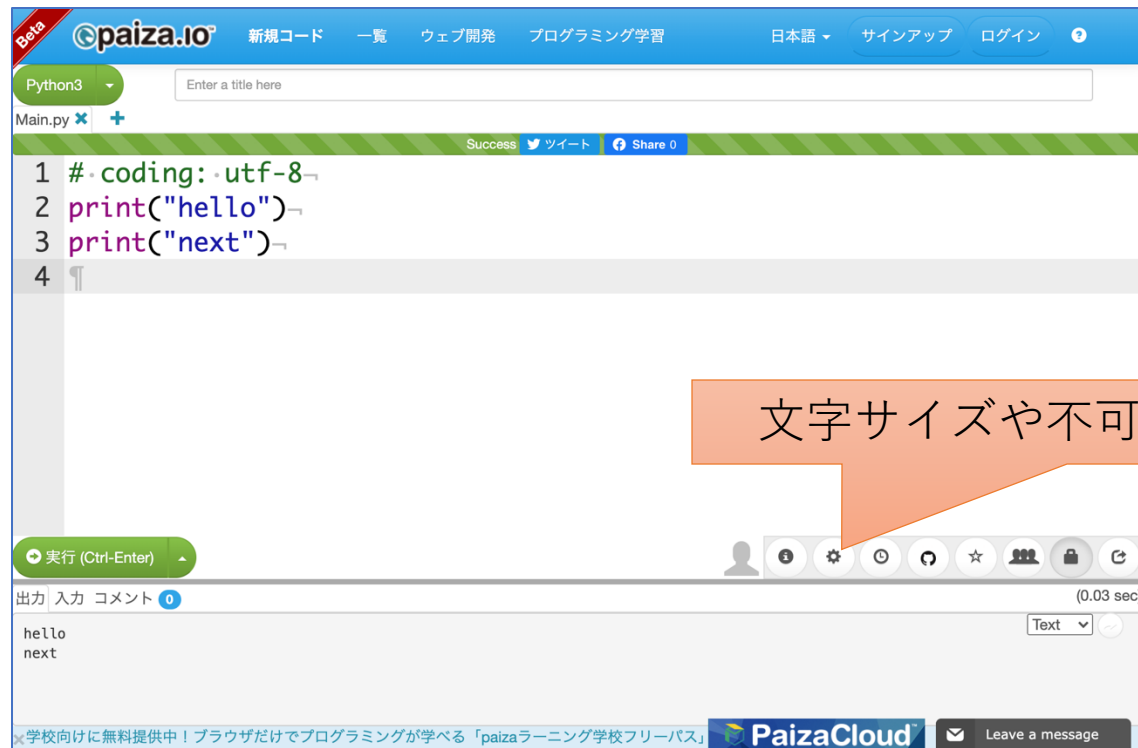


プログラミング言語の特徴

- 文法
 - 言語ごとのルール
 - キーワードの種類
- コーディング
 - プログラムの書き方
 - 開発環境（ツール）
- 性能／機能
 - 実行速度、リソース
 - マルチメディア、ネットワーク
- 得意分野
 - 利用者層（理系／文系）
 - 歴史、資産

オンライン開発環境の利用

- Paiza.io(学習専用のオンラインプログラミング環境)

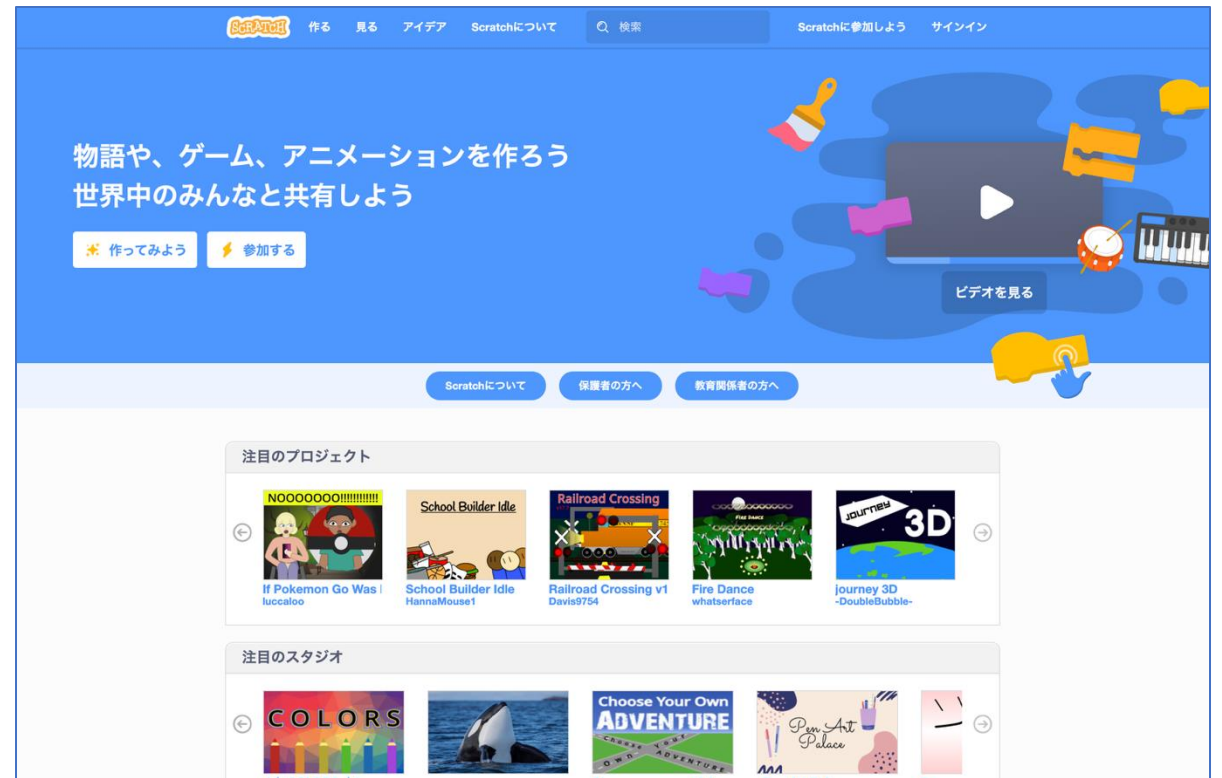


<https://paiza.io/>

プログラミング教育用言語

- Scratch (17位) [2006年登場]

- プログラミング学習環境
- ノーコードで開発できる
- オンラインコミュニティ
 - 作品公開
 - コメント
 - スタジオ
 - リミックス



「Scratch」で検索

C言語系の 3 兄弟

- C（3位）[1972年登場]
 - 高水準言語の元祖、すべてのプログラミング言語の源流
 - 低水準言語・機械語に最も近い（ハードウェアを直接操作）
- C++（2位）[1983年登場]
 - C言語の機能拡張版
 - 組み込みシステム、IoTデバイス
- C#（5位）[2002年登場]
 - C/C++の拡張版
 - Microsoft製品の中核開発言語



2022年殿堂入り

人気の言語

- Java（4位）[1995年登場]
 - C言語の正当後継者（?）
 - 元祖「オブジェクト指向」言語
 - VM(Virtual Machine)によるマルチプラットフォーム対応
- JavaScript（6位）[1995年登場]
 - Webアプリ開発
 - オブジェクト指向 + 関数型
- Python（1位）[1991年登場]
 - 少ないコード量でわかりやすく書ける
 - 科学計算、統計解析、機械学習、深層学習

Ajax + サーバサイドで
復活

特化型の言語

- R（12位）[1993年登場]
 - 統計計算、データの可視化
 - 経済学、社会学、データサイエンス
- SQL（10位）[1974年登場]
 - データベース処理に特化
 - 宣言型言語
- PHP（15位）[1995年登場]
 - Webアプリ開発に特化
 - HTMLを生成するスクリプト言語

レガシー（遺物）な言語

- COBOL（20位）[1959年登場]
 - 事務処理用、文系の人向け（？）
 - 人間の言葉（英語）に近い文法
 - 政府・企業のシステムに多数導入、移行困難で現在も稼働中
- Visual Basic（8位）[1991年登場]
 - VBA(Visual Basic for Application)の方が有名
 - Officeアプリ(Excel、Word等)がVBAをサポートするため生き残る
- LISP（24位）[1975年登場]
 - 関数型言語
 - AI研究(自然言語解析)で復権？

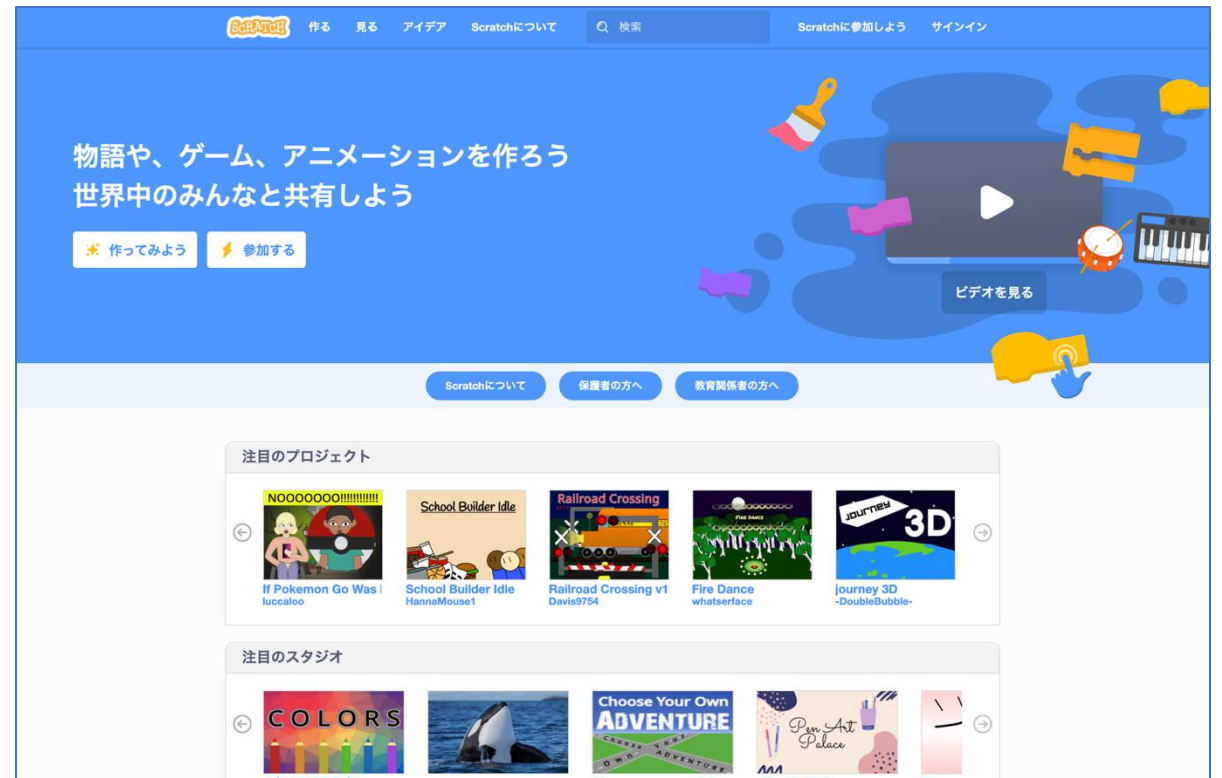
これから流行する（かも知れない）言語

- Kotlin（25位）[2011年登場]
 - Java言語の発展系、開発プロジェクトの効率化
 - Androidアプリ開発で主流になりつつある
- Go（7位）[2009年登場]
 - Google社の推奨言語
 - 軽量、高速、並行処理、シンプル
- Rust（19位）[2010年登場]
 - 新しいシステムプログラミング用言語（C/C++からの移行）
 - オープンソース、エンジニアに人気

プログラミング教育用言語

- Scratch (14位) [2006年登場]

- プログラミング学習環境
- ノーコードで開発できる
- オンラインコミュニティ
 - 作品公開
 - コメント
 - スタジオ
 - リミックス



「Scratch」で検索

プログラミング教育用言語

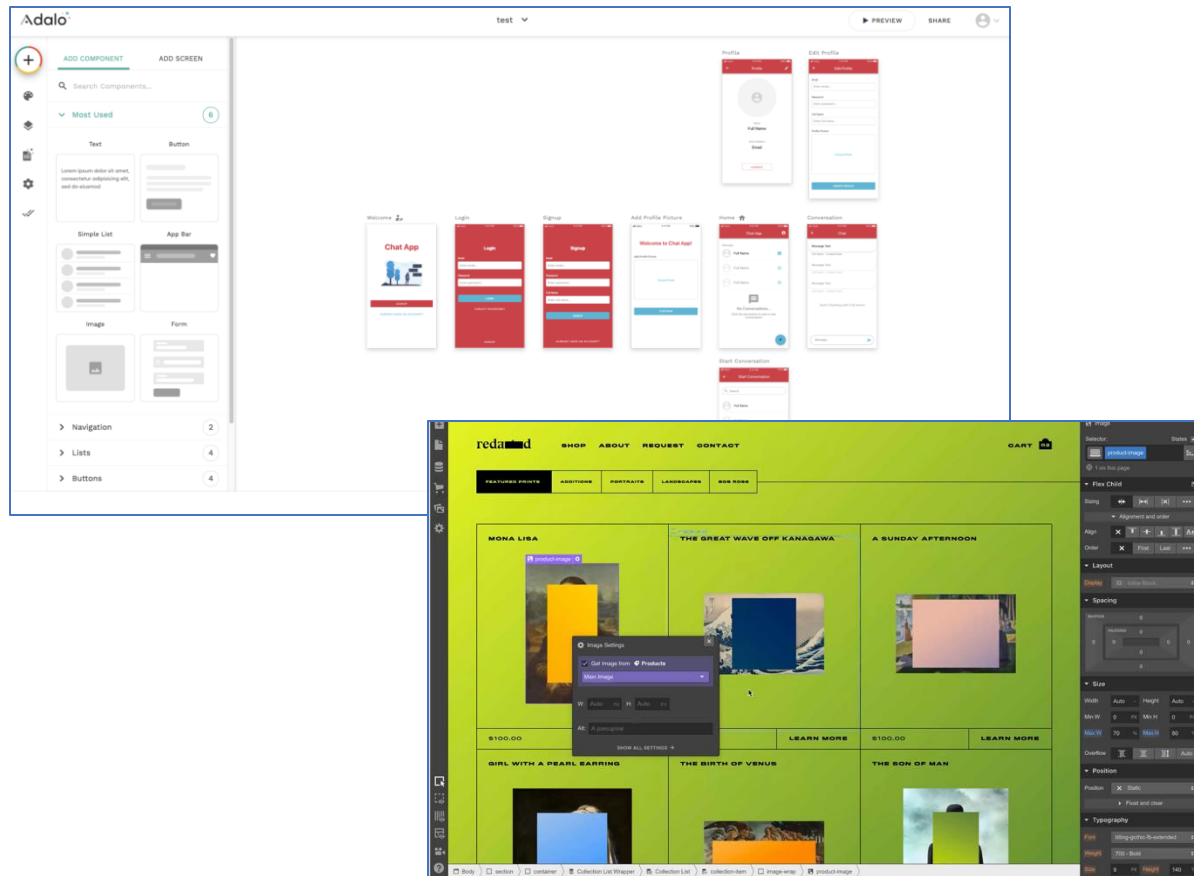
- アルゴロジック
 - 電子情報技術産業協会（JEITA）
 - アルゴリズムの学習特化型
 - 学習記録、チャレンジ問題



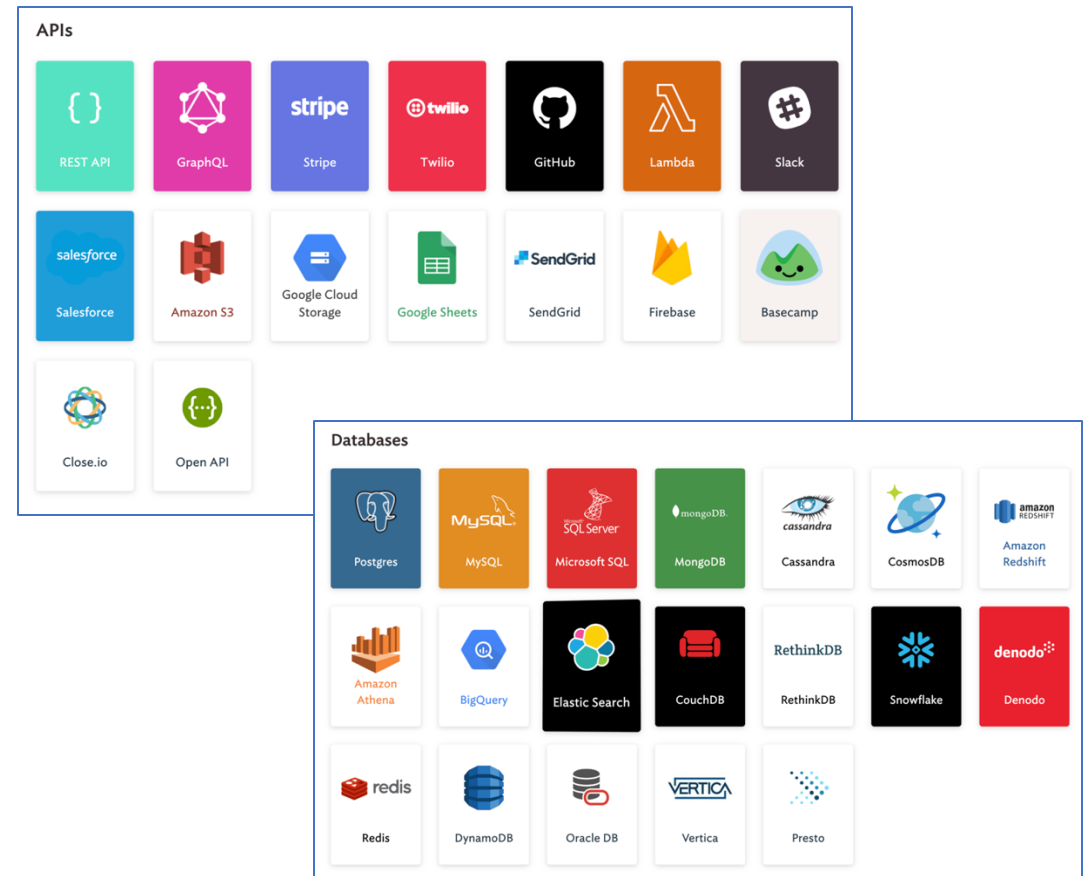
「アルゴロジック」で検索

ローコード、ノーコード

• Adalo/Webflow/Bubble ...

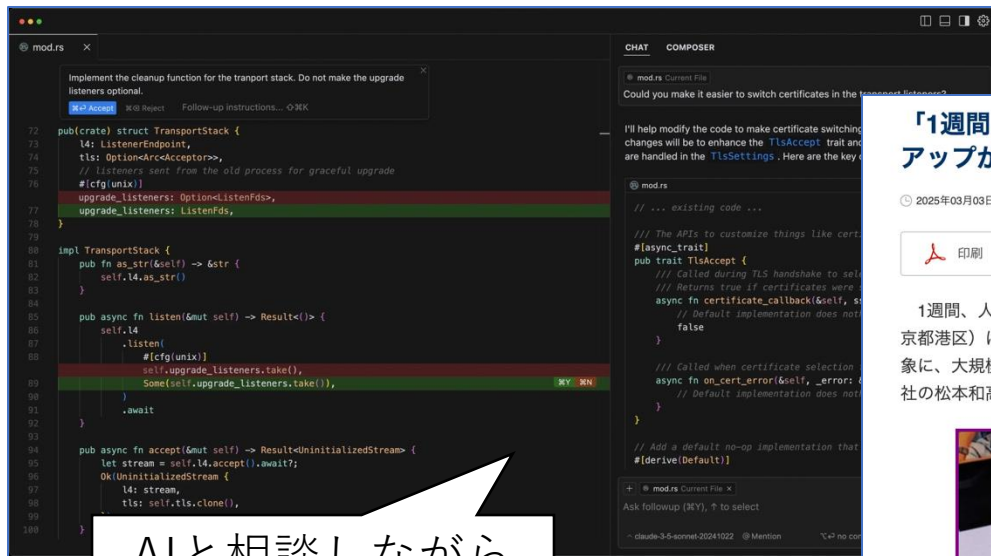


• Retool/Monaca/Mendix ...



ノーコードの極致

• 生成AIによるプログラミング



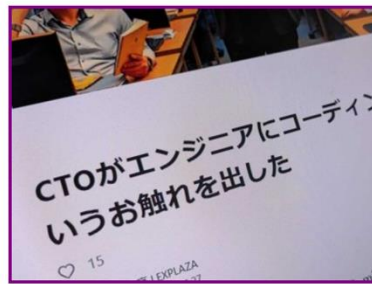
AIと相談しながら
コーディング

「1週間、人の手でコーディング禁止」——AIスタートアップが社内検証 AI生成コードのみ利用OK

2025年03月03日 17時43分 公開

[松浦立樹, ITmedia]

1週間、人の手でコーディング禁止——AIスタートアップ企業のエクスペリエンス。東京都港区は3月3日、こんな実験を始めると発表した。同社所属のエンジニアに、大規模言語モデル（LLM）を使ったコーディングのみを許可すると、同社の松本和高CTOが自身のnoteでその経緯などを説明している。



「1週間、人の手でコーディング禁止」——AIスタートアップが社

NASAの定めるコード規約

The Power of 10: Rules for Developing Safety-Critical Code

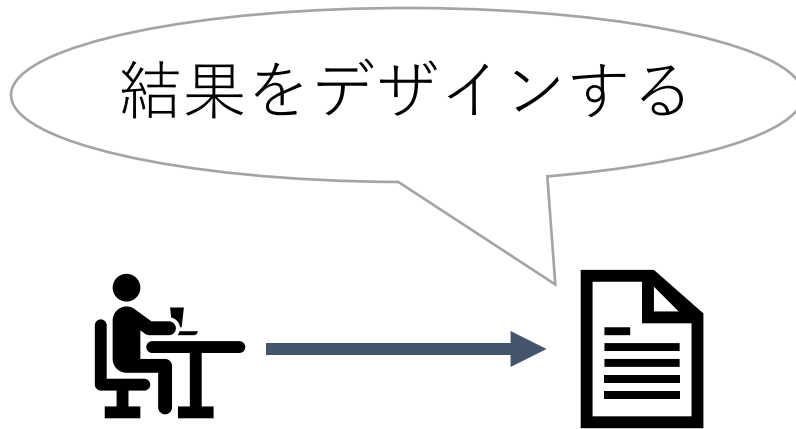
Rules [edit]

The ten rules are:^[1]

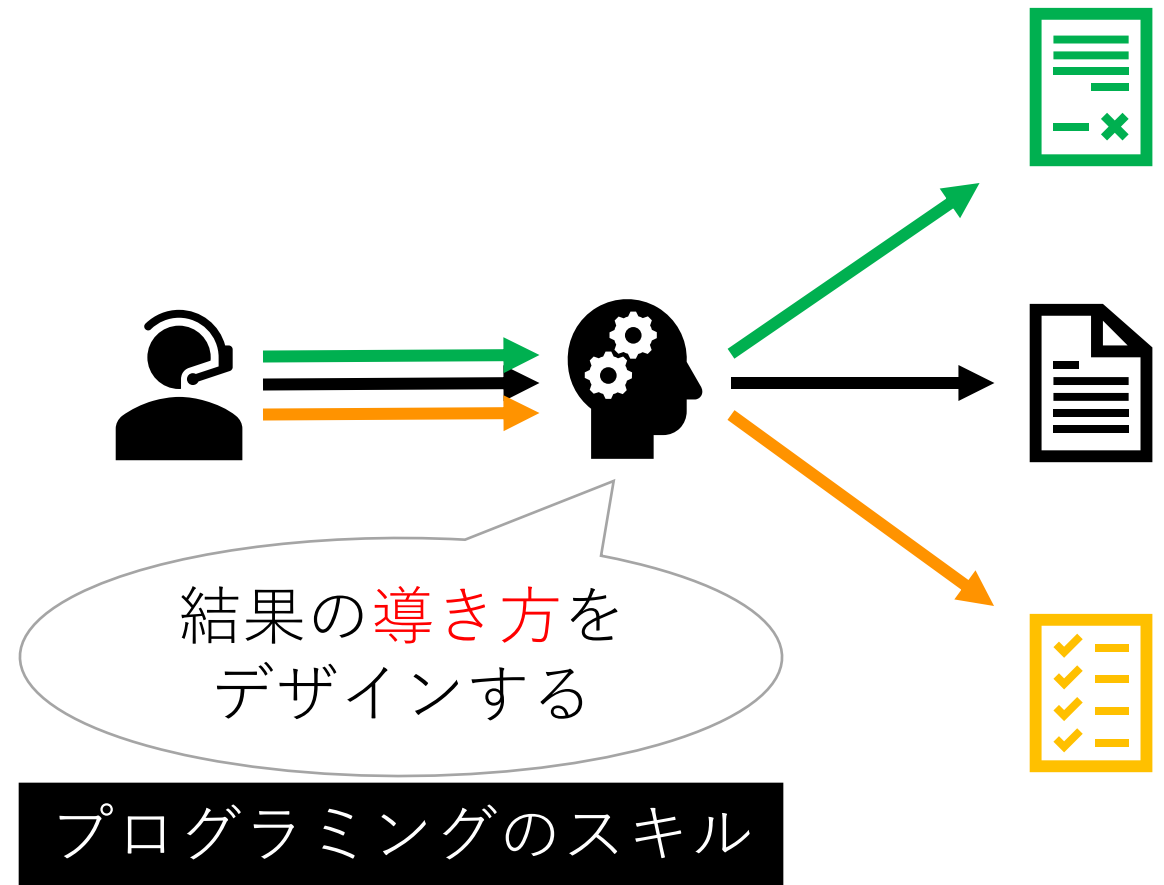
1. Avoid complex flow constructs, such as [goto](#) and [recursion](#).
2. All loops must have fixed bounds. This prevents runaway code.
3. Avoid [heap memory allocation](#) after [initialization](#).
4. Restrict functions to a single printed page.
5. Use a minimum of two [runtime assertions](#) per function.
6. Restrict the scope of data to the smallest possible.
7. Check the return value of all non-void functions, or cast to void to indicate the return value is useless.
8. Use the [preprocessor](#) only for [header files](#) and simple [macros](#).
9. Limit pointer use to a single [dereference](#), and do not use [function pointers](#).
10. Compile with all possible warnings active; all warnings should then be addressed before release of the software.

AI時代の働き方

- これまで（AI以前）



- これから（AI以後）



プログラミングの学び方

- 学校なんか必要ない？



The screenshot shows the homepage of 'ドットインストール PREMIUM' (Dots Install Premium). The header includes navigation links like 'マイページ', 'レッスン一覧', and 'プレミアム会員'. A main banner features the text '3分動画でマスターできる プログラミング学習サービス' (Master programming with 3-minute videos) and '451 レッスン を 6,777 本の動画 にて提供中' (Provided in 6,777 videos for 451 lessons). Below this is a blue button 'すべてのレッスンを見る'. The '人気のレッスン' (Popular Lessons) section lists several courses with icons and descriptions:

- 実践！アプリ紹介ページを作ろう (全16回)**: スマートフォン画面にも対応したアプリ紹介ページを制作しながら、HTML/CSSの実践的なテクニックを学んでいきます。
- はじめてのJavaScript (全11回)**: ブラウザを操作することができるJavaScriptについて、簡単なゲームを作りながら基礎を学んでいきます。
- 詳解PHP 基礎文法編 (全34回)**: Webサービス開発でよく使われるPHPの基礎的な文法について学んでいきます。
- はじめてのPython (全9回)**: シンプルなコードでプログラムを書いていくことができるPythonについて、簡単なゲームを作りながら基礎を学んでいきます。
- はじめてのRuby (全9回)**: 短いコードで直感的に書いていくことができ
- Unity入門 (全26回)**: 強力なゲームエンジンであるUnityの使い方



The screenshot shows the homepage of 'paizaラーニング'. The header includes navigation links like '総合トップ', 'エンジニア転職', and 'プログラミング学習'. A main banner features the text '1本3分の動画と練習問題で 効率的に学ぶ' (Learn efficiently with 1 video and 3 minutes of practice problems). Below this is a yellow button '学習を始める (会員登録)' (Start learning (register)). A video player is shown with a play button and a progress bar. The 'paizaラーニングの特徴' (Features of paiza Learning) section lists several benefits:

- 環境構築不要5秒で開始！**: 実行環境構築いらずでブラウザ上でプログラミング言語を実行できるので、誰でもすぐに学習を始められます。
- 3分動画で毎日続けられる**: 動画レッスンを見ながらプログラミングすることで理解を深め、その後に練習問題を解いて応用力をきたえることができます。
- 不明点はエンジニアに質問**: 弊社エンジニアに質問をして、アドバイスをもらうことが可能。初歩的なところでもまずいて、やる気を失ってしまうことはありません。

プログラミングをはじめよう！

- プログラミングを学ぶ意義

- 論理的な思考力を育む
- デザインセンスを育む
- 表現力を育む

- プログラミングの楽しさ

- 作ったものが動く楽しさ（ものづくり）
- 身の回りのあらゆるものを動かせる（ユビキタス）
- 世界中のものを動かせる（インターネット）
- 現実以上のものも作れる（バーチャルリアリティ）



世界中で、子供から大人まで、
すべての人が