Monaca

プログラミング教育オンライン研修会

～文科省教員研修用教材(JavaScript版)のポイントを踏まえたプログラミング入門～

アシアル株式会社
アシアル情報教育研究所
岡本 雄樹

自己紹介

■名前

- 岡本雄樹(アシアル情報教育研究所 所長)

■著書

- イラストでよくわかるPHP
- WordPressプロフェッショナル養成読本
- Monacaで学ぶはじめてのプログラミング

■メッセージ

- 「コンピューター」「インターネット」「プログラミング」
- 私は高校生の時にそれらと出会うことで人生が拓けました。
- 先生方とMonacaによるアプリ開発を通じて、情報技術の活用方法や作品作りの楽しさを広めてまいります。

アシアル株式会社について

- 2002年
  - 代表の田中正裕が本郷の地で創業（当時20才）
    - PHP言語に関する雑誌発刊・教育・コンサルティング
- 2010年
  - アシアルPHPスクールのマネージャーに岡本が就任
- 2012年
  - アプリ開発ツール「Monaca」をリリース
    - JavaScript言語とHTML5による複数OS向けアプリ開発環境を提供
- 2015年
  - MonacaEducation事業がスタート
- 2020年
  - アシアル情報教育研究所設立

アシアル株式会社

会社概要

- 創業：2002年
- 事業所：東京（本社）、サンフランシスコ、ハンガリー
- 従業員数：約60名（10国籍）

社名の由来

- アジアのリーダー

事業内容

- システム構築・アプリ開発
- 製品・サービス
- プログラミング教育

新指導要領について



新学習指導要領とプログラミング

- 小学校(2020～)
  - 既存の教科の中で実施
- 中学校(2021～)
  - 技術・家庭科の中で実施
- 高校(2022～)
  - 教科「情報」で実施
    - 共通必履修科目「情報Ⅰ」が2単位（70時間）

中学校のプログラミング教育

- 現行
  - プログラムによる計測・制御
- 次期
  - プログラムによる計測・制御
  - 「ネットワークを利用した双方向性のあるコンテンツのプログラミング」

高校のプログラミング教育

- 現行
  - 教科「情報」は2003年度から存在する
  - 現行科目は「社会と情報」と「情報の科学」の選択必履修
  - プログラミングを扱うのは「情報の科学」
  - 「情報の科学」は2割程度の学校が履修
- 次期
  - 新科目「情報Ⅰ」に一本化されプログラミングが必ず行われる
  - 更に新科目「情報Ⅱ」が選択科目として追加される
  - 「総合的な探究の時間」で行われる場合も
  - 入試科目に！

情報Ⅰ

情報に関する科学的な見方・考え方を働かせ，情報技術を活用して問題の発見・解決を行う学習活動を通して，問題の発見・解決に向けて情報と情報技術を適切かつ効果的に活用し，情報社会に主体的に参画するための資質・能力を次のとおり育成することを目指す。

(1) 情報社会の問題解決
(2) コミュニケーションと情報デザイン
(3) コンピュータとプログラミング
(4) 情報通信ネットワークとデータの活用

改訂のポイントは問題の発見と解決

文科省の前調査官によると改訂のポイントは
`問題の発見と解決`

やりたいことを見つけ実現する力を養う、“やりたいことができる教育”に。

学びの場.com　情報科の改訂のポイントを教えていただけますか？

鹿野 利春　“問題の発見と解決”がポイントです。情報というと技術的なことを想像されている方が多いかもしれませんが、情報をデザインしたりプログラミングしたりデータを活用したりすることは手段であって、プログラミングの技術を磨くことがねらいではありません。情報科改訂のねらいは問題を発見することと、どうすればよいのか解決策を考えること、この両方の思考を養うことにあります。その解決策はプログラミングを必要としないかもしれませんが、プログラミングを使えれば効率的に解決することが可能となるでしょう。

学びの場.com　具体的にどんなふうに学んでいくことを想定されていますか？

鹿野 利春　例えば「スマホと人でしりとりができるようになりたいな」といったふうに、やりたいことを見つけて取り組むことを想定しています。社会問題などを取り扱ってもいいですが、もっと身近な問題でいいのです。もちろん先生のリード次第ですが、「スマホで写真に落書きをするアプリを作りたい」でも。自分で興味のあることや、何を実現したいのかを見つけて、そのために何をしたらいいのかを自分自身で見つけることが大切です。

鹿野利春　高等学校の「情報科」改訂を語る。 - 教育インタビュー ｜ 学びの場.com
https://www.manabinoba.com/interview/018188.html

JavaScript入門



Monaca

幾つかの言語と成り立ち

- **BASIC言語**
  - Beginner向けの言語として1964年に登場
- **C言語**
  - 主にOS（UNIX）開発のため1970年代初頭に登場
- **VBA**
  - MS Office上で動作する言語として1993年に登場
- **JavaScript**
  - ブラウザ上で動くプログラミング言語として1995年に登場

言語も、何らかの課題を解決するために登場しています

JavaScriptは何故生まれたか？

- ブラウザ上でプログラムを動かしたかったから
  - HTML言語だけではコンテンツ(画像や表など)の表示しかできない
    - JavaScriptがあれば画面に動きをつけることができる
  - 現在のWebは、JavaScriptが外部と通信することで(WebAPI)高度なサービスの提供を実現している
    - 初期の代表例はGoogle Maps
    - 今年度、気象庁のWebサイトがWebAPIベースに置き換えられ、高機能化
  - ブラウザさえあればOSは何でも良い、という世界観も
    - そしてChrome OSが誕生

JavaScriptの存在意義

- クロスプラットフォーム
  - ブラウザさえあればどのOSでも利用できる
    - 言語の文法も「Ecma」という標準化団体が標準化済み
    - 後方互換性も維持されている

教育現場におけるJavaScriptのメリット

- 生徒の反応が良い
  - 画像や音を簡単に扱える
- 学校の外でも気軽に取り組める
  - ブラウザさえあれば動かせる
  - 生徒が自宅で予習や復習しやすい
  - 研修もブラウザベースの方が組みやすい
- 教材の寿命が長い
  - 指導案やプリントを使い続けられる

文科省教員研修用教材・第3章
「コンピュータとプログラミング」ポイント解説

Monaca

第三章の内容

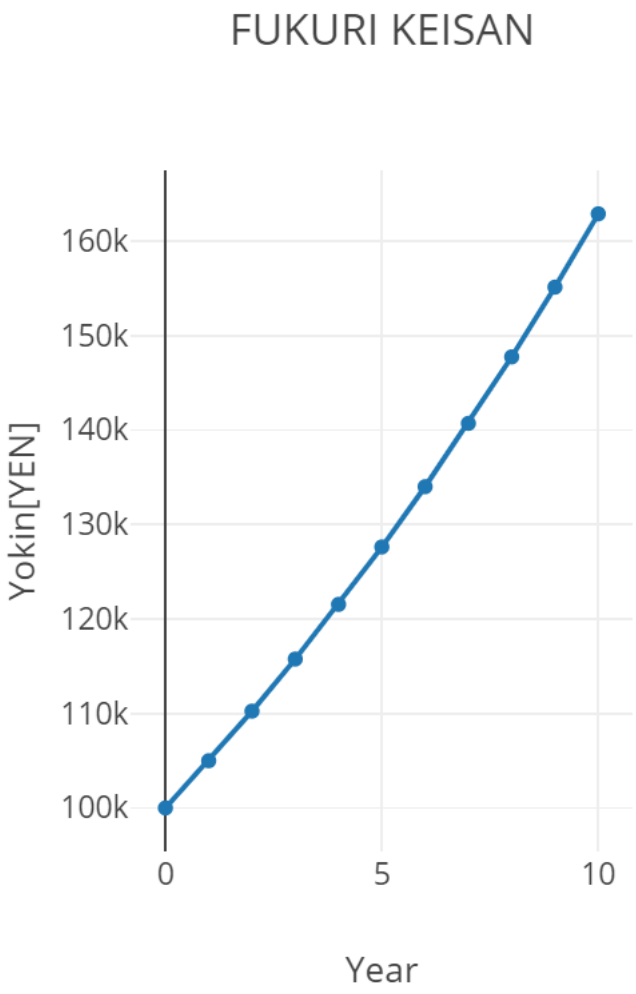
学習	タイトル	ポイント
1 1	コンピュータの仕組み	論理回路 桁あふれ
1 2	外部装置との接続	micro:bitなど 順次・分岐・繰り返し
1 3	基本的プログラム	変数・順次・分岐・繰り返し
1 4	応用的プログラム	関数・乱数・配列・WebAPI
1 5	アルゴリズムの比較	複数の分岐と繰り返しが登場
1 6	確定モデルと確率モデル	グラフ
1 7	自然現象のモデル化とシミュレーション	グラフ

学習の基本方針

- Lv0
  - コンピューターの仕組みを理解する
- Lv1
  - 順次・分岐・繰り返しを理解して応用できるようにする
- Lv2
  - 配列と関数を理解して応用できるようにする
- Lv3
  - グラフや代表的なアルゴリズムのプログラムを読めるようにする

応用を学ぶには配列が重要！

複利計算のグラフ



その時の配列イメージ

[9]	162889
[8]	155132
[7]	147745
[6]	140710
[5]	134009
[4]	127628
[3]	121550
[2]	115762
[1]	110250
[0]	100000

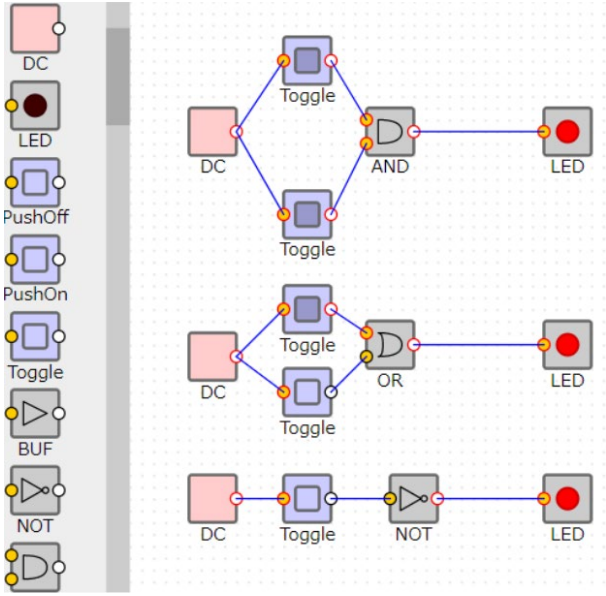
[キー] バリュー

Webで学ぶ論理回路

Monaca

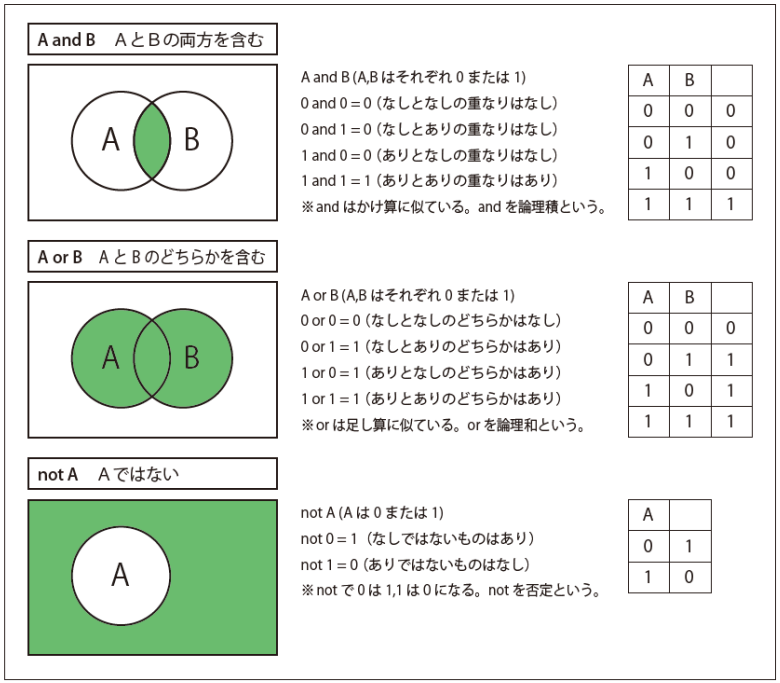
【実習】AND回路とOR回路とNOT回路体験

- **論理回路シミュレータ SimcirJS**
  - ブラウザ上で論理回路の組み立てを行える
  - ライセンスはオープンソース(MIT)
  - 本実習で必要な装置
    - DC(入力)
    - Toggle(on/off)
    - AND
    - OR
    - NOT



【実習】AND回路とOR回路とNOT回路体験

- 教員研修資料の挙動を確認してみよう



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

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

A	
0	1
1	0

図表 4　論理演算

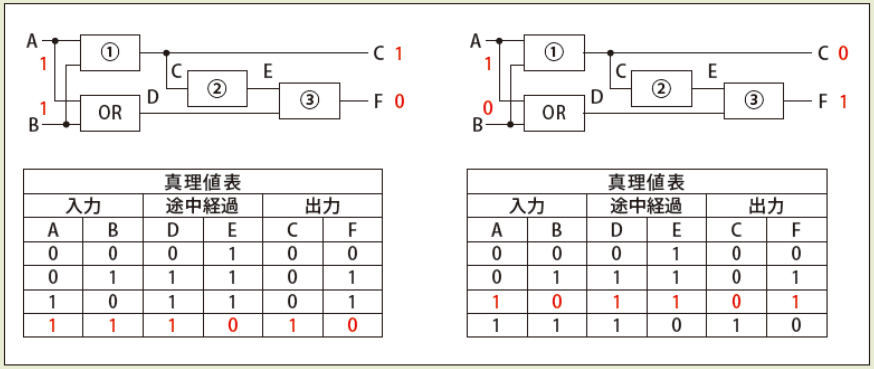
※高等学校情報科「情報Ⅰ」教員研修用教材 JavaScript版 p5

【実習】AND回路とOR回路とNOT回路体験

- 演習2で示されている「二進数の足し算」に挑戦

＜演習 2＞

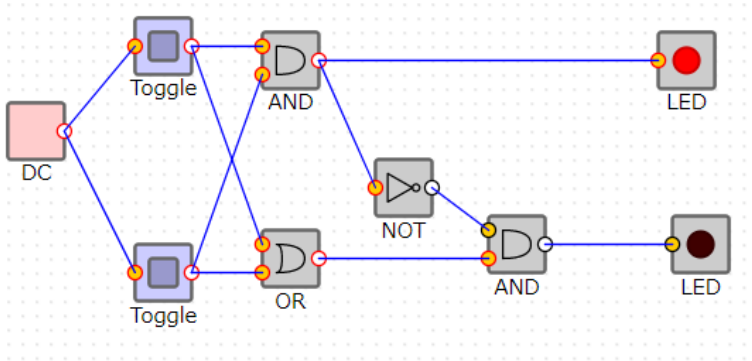
次の論理回路は入力のAとBがともに1の場合，出力のCが1，Fが0となります。また入力のAが1，Bが0の場合，出力のCが0，Fが1となるものです。この論理回路の①，②，③に適切な論理演算を入れ，上記の二進数の足し算を成立させましょう。



真理値表					
入力		途中経過		出力	
A	B	D	E	C	F
0	0	0	1	0	0
0	1	1	1	0	1
1	0	1	1	0	1
1	1	1	0	1	0

真理値表					
入力		途中経過		出力	
A	B	D	E	C	F
0	0	0	1	0	0
0	1	1	1	0	1
1	0	1	1	0	1
1	1	1	0	1	0

図表 5　論理回路と真理値



※高等学校情報科「情報Ⅰ」教員研修用教材 JavaScript版 p6

Webで学ぶ外部装置

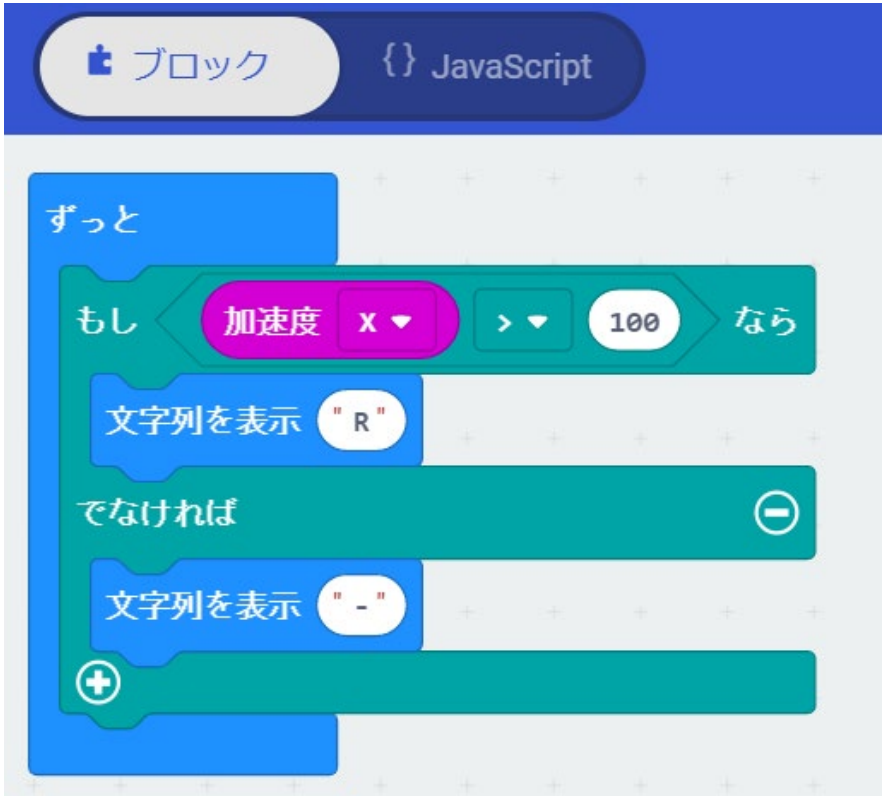
【実習】外部装置

- MicrosoftMakeCode
  - ブラウザ上でMicro:bitのプログラミングができる
  - エミュレータも付いている
  - JavaScriptとブロックの相互変換も可能

【実習】外部装置で分岐と反復

分岐

繰り返し



【参考】分岐と反復のJavaScriptソースコード

分岐

```
basic.forever(function () {
    if (input.acceleration(Dimension.X) > 100) {
        basic.showString("R")
    } else {
       basic.showString("-")
    }
})
```

繰り返し

```
let i = 0
for (let i0 = 0; i0 < 10; i0++) {
    i += 1
    basic.showString("" + i)
    basic.pause(1000)
    basic.clearScreen()
    basic.pause(1000)
}
```

※高等学校情報科「情報Ⅰ」教員研修用教材 JavaScript版 p.6-7

if文とfor文では【条件式】を利用している

繰り返し(while文)の紹介

for文から条件式以外を取り除けばwhile文に！

フローチャート

■ フローチャート(流れ図)

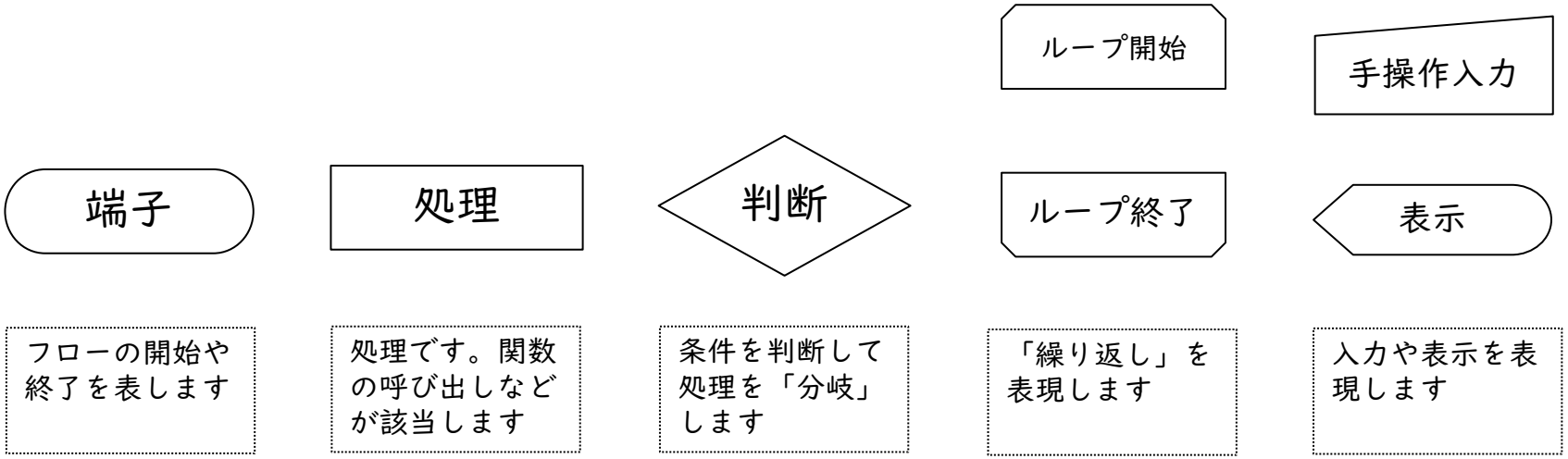
- 制御構造はフローチャートで図式化できます
  - 基本的には上から下へ、左から右へ流れていきます
  - 元々は工学や経営のために誕生したツールです
  - 日本ではJISで規格化もされています
- 高級言語が登場するもはるか昔に作られた図法のため、表現力には限界があります。
  - そのため、実際のプログラミングではあまり使われていません。

■ フローチャートの書き方(ツール編)

- 作図はソフトの利用をお勧めします
  - 図形と線の接続・移動が簡単
- フローチャートは汎用ソフトで対応可能
  - draw.ioのような汎用的な作図ツールでも十分、描けます
  - プレゼンテーションソフトにもフローチャートの図形は入っています
- プログラミング用の様々な図について
  - 専用の作図ツールも存在します

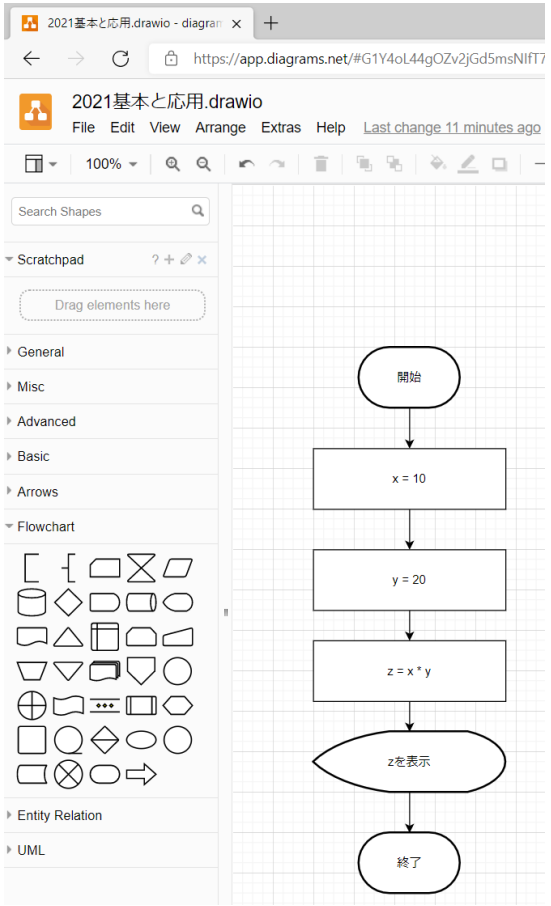
■ フローチャートの書き方(記号編)

- 主要な記号を紹介します
- 実は「判断」でループを表現することも可能です
- 入力や表示を全部「処理」記号で表しても表現は可能です



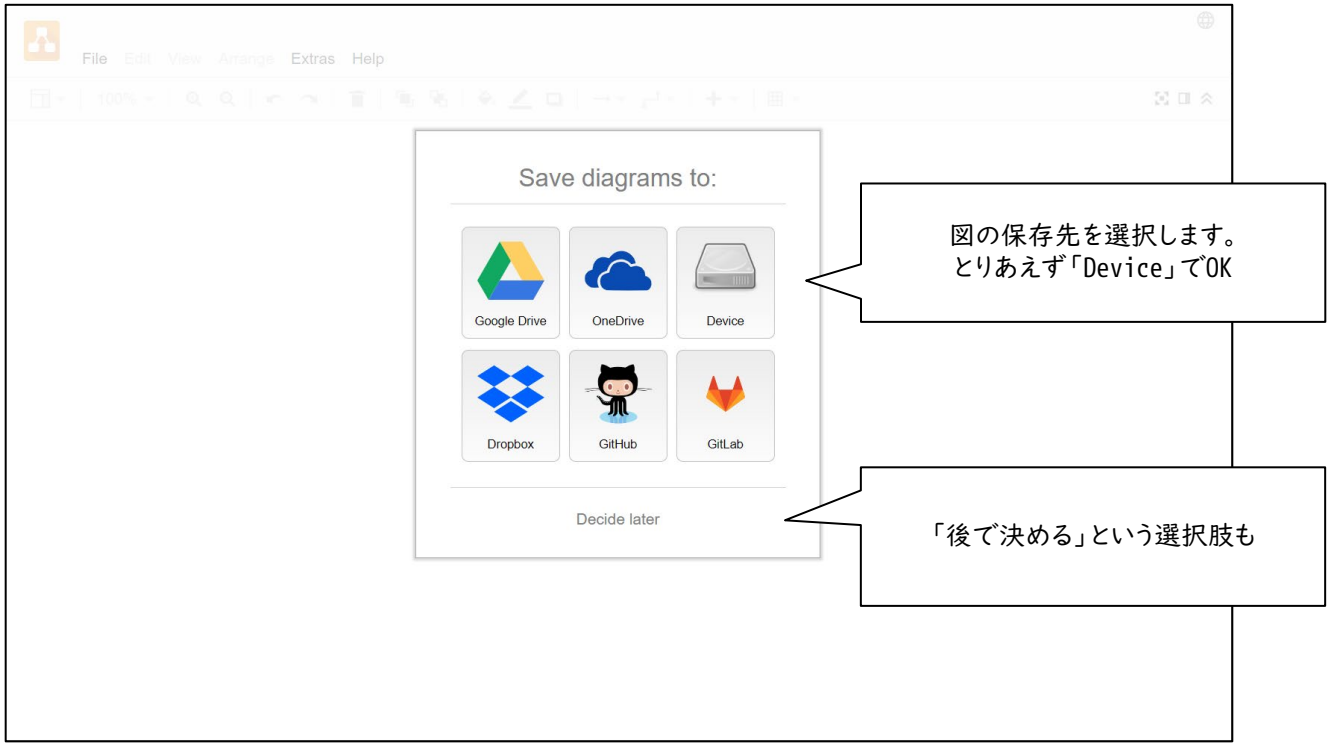
draw.io(diagrams.net)による作図入門

- 作図ツールを使うと部品や線の配置がとても楽です
- 研修用教材で紹介されているdraw.ioを紹介します
- ブラウザ上で動作します
- 会員登録・ログイン不要です



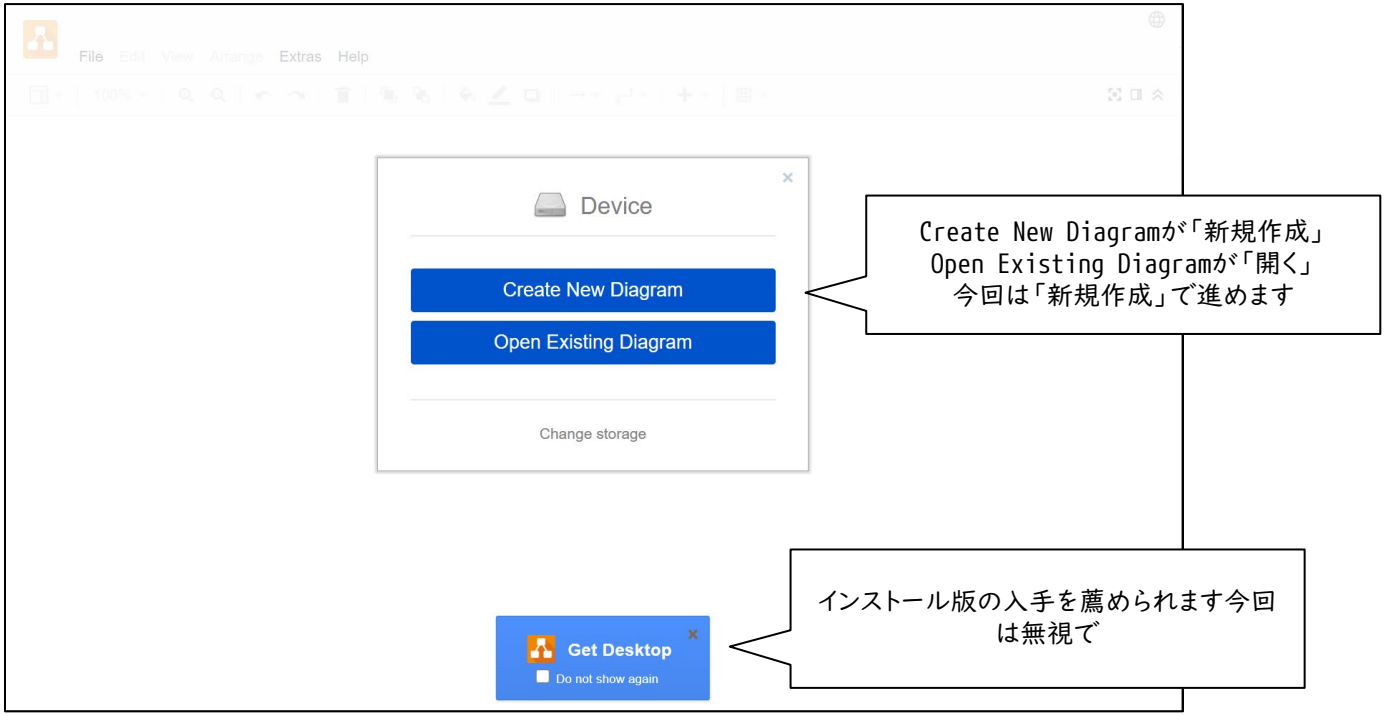
draw.ioの起動

- https://app.diagrams.net/ にアクセス
  - 旧URLは https://draw.io でした



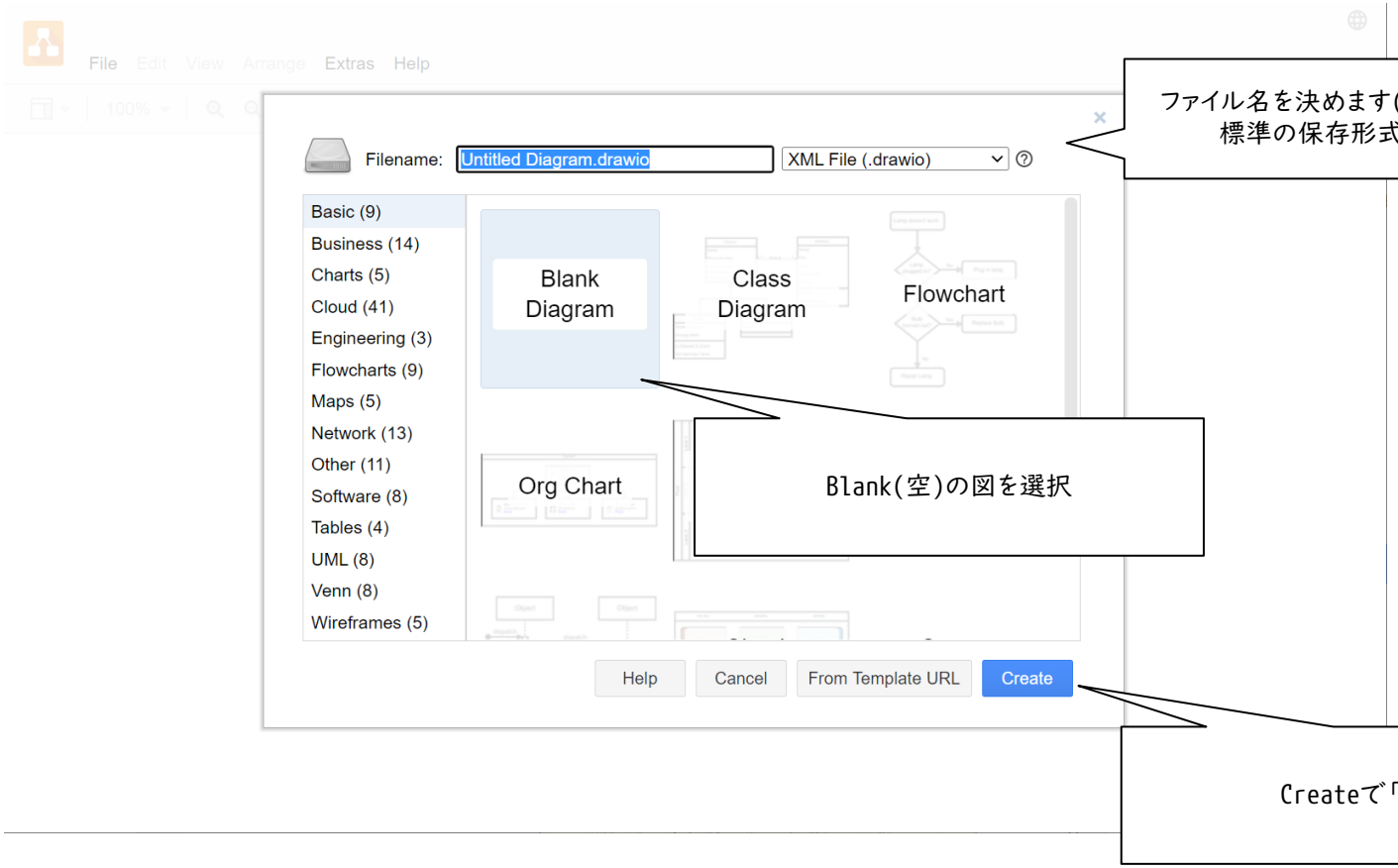
draw.ioの起動

「新規作成」か既存ファイルを「開く」のか選択します



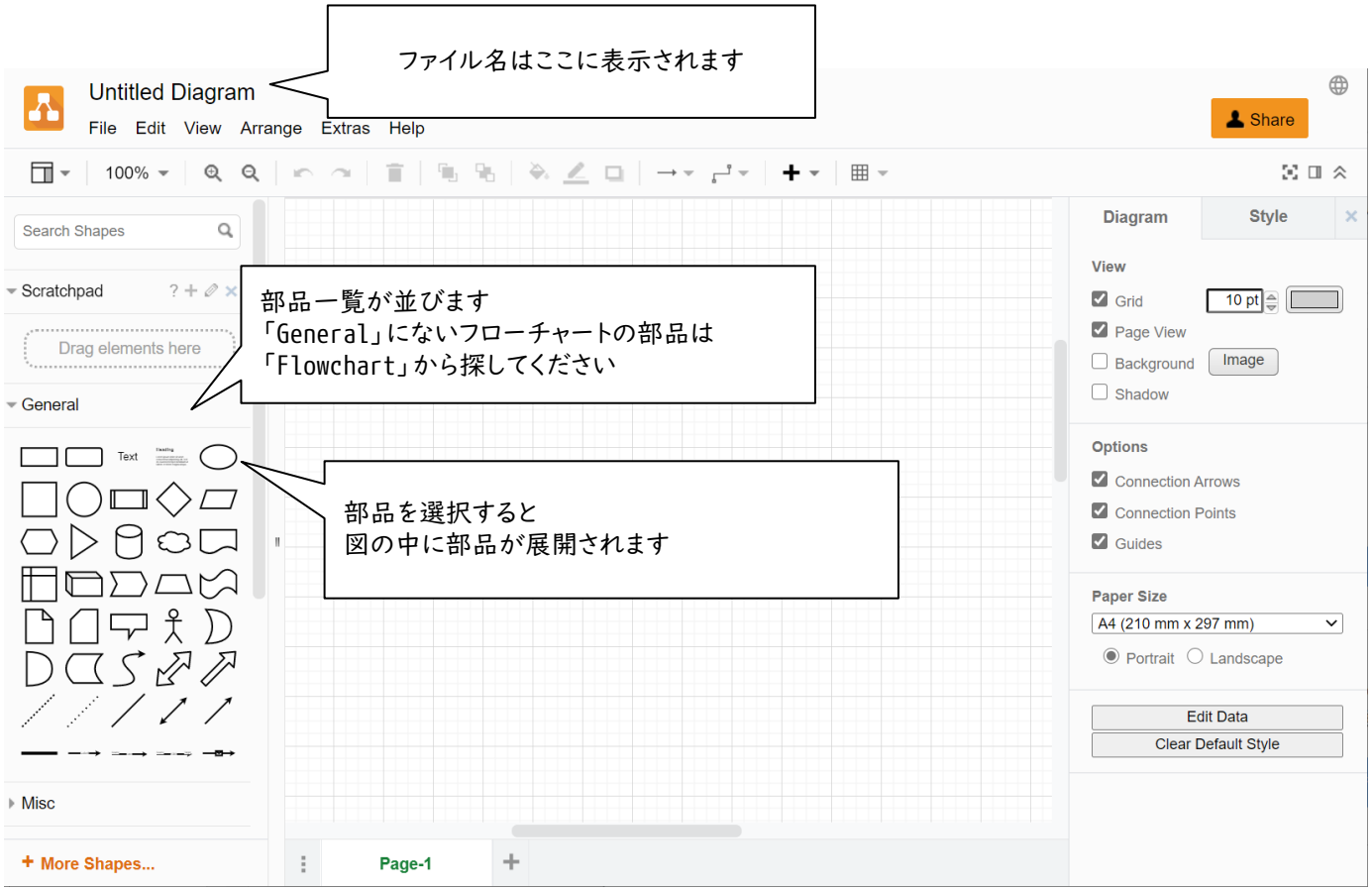
draw.ioの起動

- ファイル名とテンプレートを選んで新規作成を行います



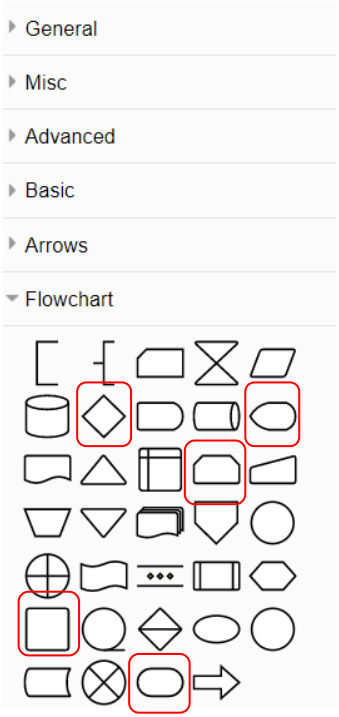
draw.ioの起動

- 起動したらまずはフローチャート用の部品を探しましょう

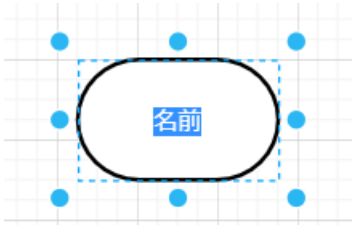


draw.ioの起動

- フローチャート用の部品は『Flowchart』としてまとまっています
- 部品は図に展開後、ラベル名を付けることができます
  - 『ダブルクリック』もしくは『F2』キーで編集可能



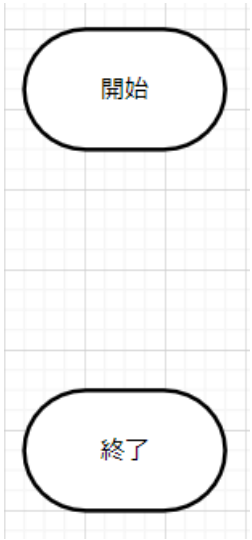
フローチャートの部品



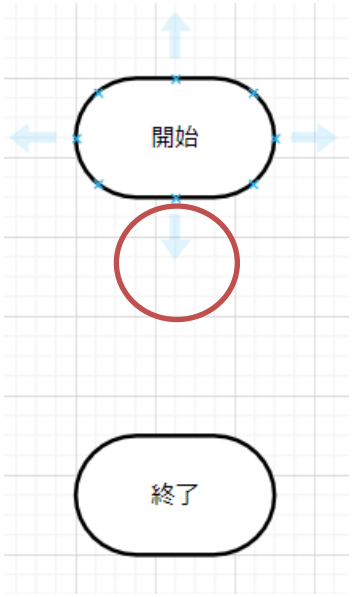
ラベル名の変更

■ 【実習】draw.ioの起動と基本操作

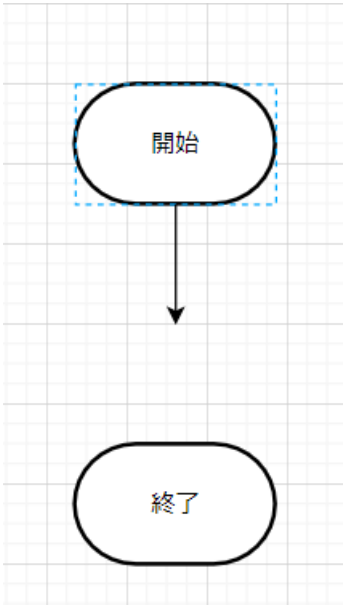
- 部品同士を線で結んでみましょう
  - 端子を二つ用意して線で結びます
  - ラベル名も変えてみましょう



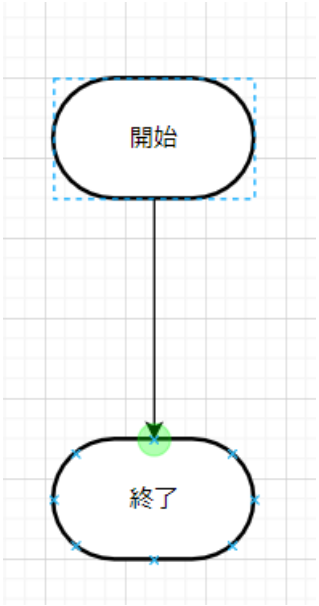
①部品を二つ用意します



②マウスカーソルをホバーさせ、矢印をクリック



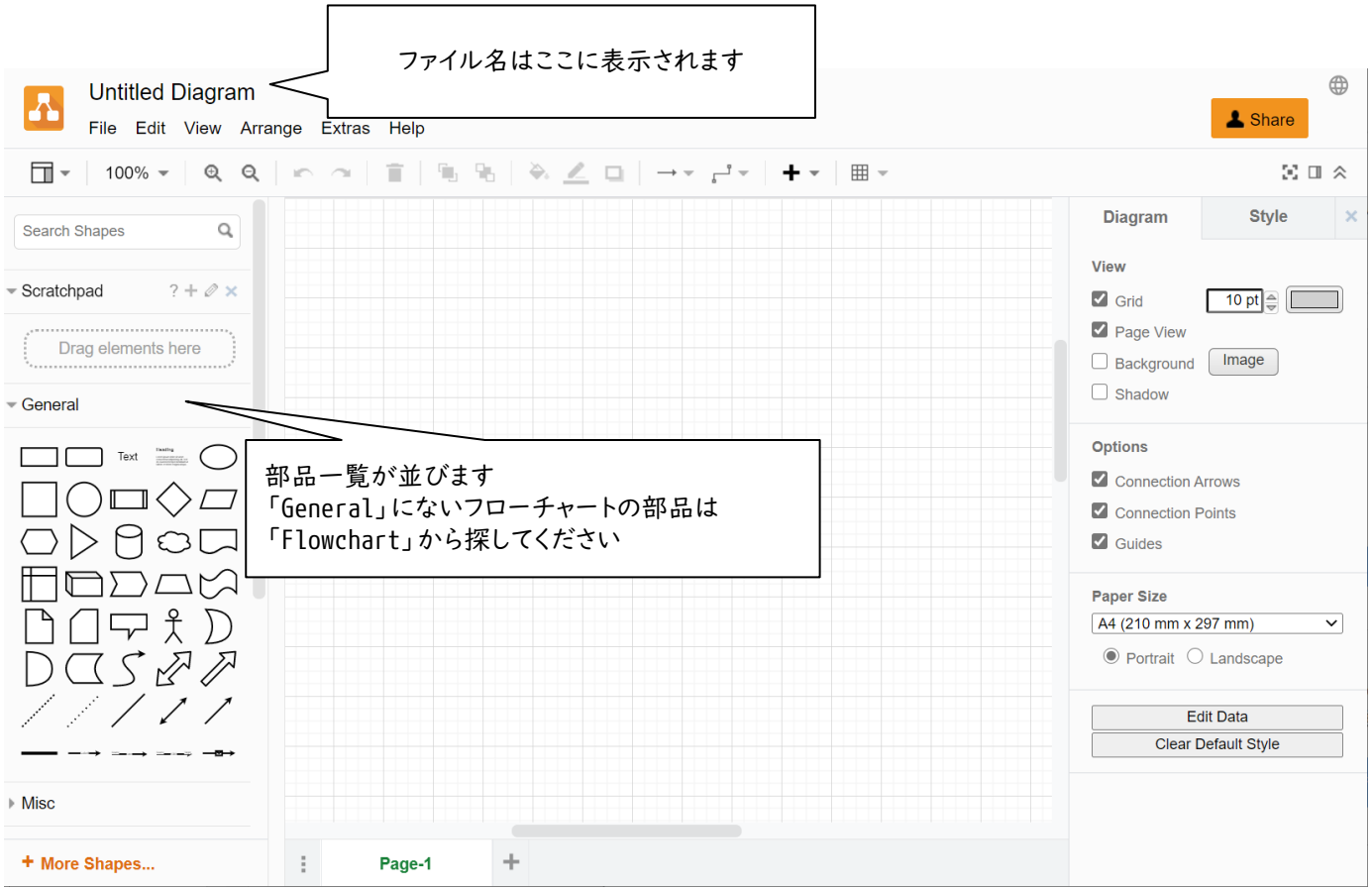
③部品から線が出ます。



④矢印の先を部品に接続して線を接続します。

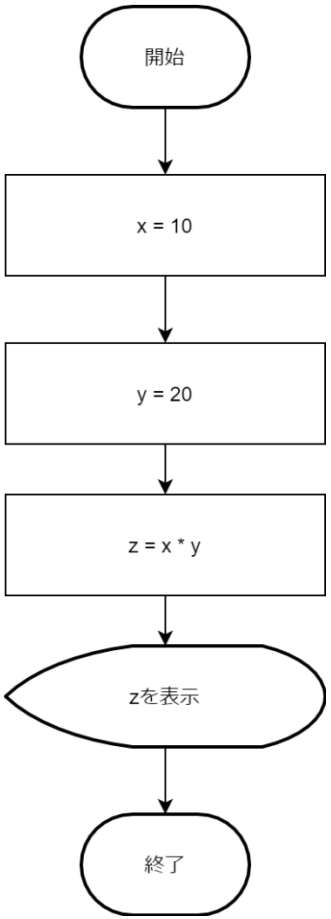
draw.ioの起動

起動したらまずはフローチャート用の部品を探しましょう



■ 順次構造の作成

└ 図のような順次構造を作図してみましょう



サンプルコード

```
x = 10;
y = 20;
z = x * y;

document.write(z);
```

■ 疑似言語(文科省教材版)の対応表

- 文科省教材ではフローチャートに疑似言語が使用されています
  - 疑似言語の書き方は自由
  - 疑似言語の仕様の一つとして、DNCL(センター試験用手順記述標準言語)というものもあります

	疑似言語	JavaScript
変数に値を代入	x ← 0	x = 0;

変数の値を10増やす	x ← x + 10	x += 10;
		もしくは
		x = x + 10;

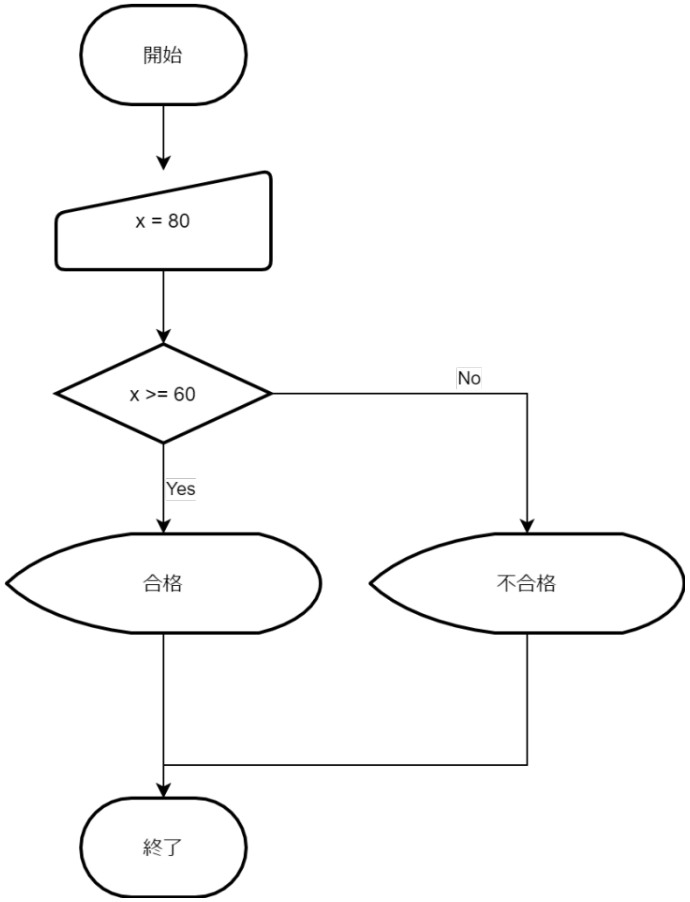
変数の値が60以上か	x ≧ 60	if (x >= 60)
奇数かどうか	i % 2 == 1	if (i % 2 == 1)

値を表示	x	document.write(x);
		もしくは
		console.log(x);
		alert(x);

繰り返し	i ← 1,2,3,4,5	for(i = 1; i < 6; i++)

■ 分岐の作成

└ 図のような分岐構造を作図してみましょう



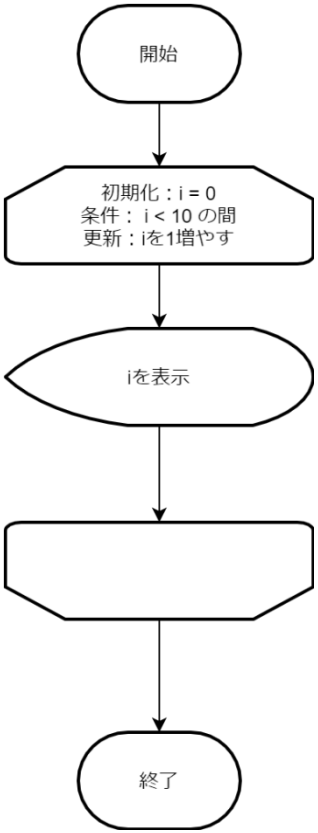
サンプルコード

```
x = 80;

if (x >= 60) {
    document.write("合格");
} else {
    document.write("不合格");
}
```

■ 繰り返しの作成

└ 図のような繰り返し構造を作図してみましょう



サンプルコード

```
for (i = 0; i < 10; i++) {
    document.write(i);
}
```

実行結果

```
0 1 2 3 4 5 6 7 8 9
```

解説

for文で「更新」が実施されるのは
ループ終端のタイミングです。
iの値が10になった次の回のループは条件が不一致となるため実行されず終了となります。