

目次

序章	Monaca Education の利用方法	2
第1章	順次実行と変数	10
第2章	条件分岐による選択（分岐）	20
第3章	配列	34
第4章	繰り返し（反復）	42
第5章	関数の定義と利用	52
第6章	繰り返し（反復）と選択（分岐）の組み合わせ	62
付録	DOM の操作	72

序章 Monaca Education の利用方法

Monaca Education を利用するためにはアカウントが必要です。先生の指示に従ってアカウントを事前に準備して下さい。

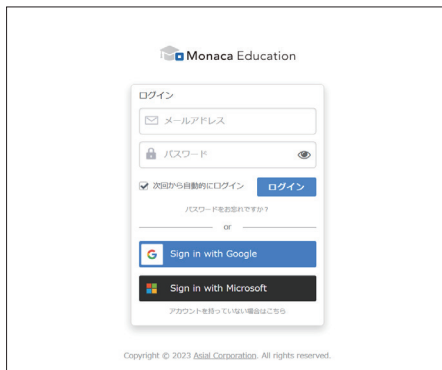
Monaca Education にログインする

Monaca Education の公式サイトにアクセスします。

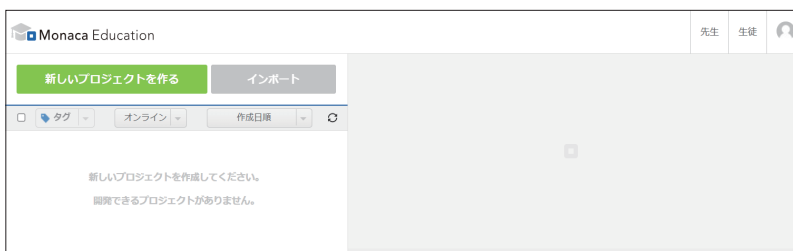
<https://edu.monaca.io>



次に右上の「ログイン」をクリックして下さい。ログインフォームが表示されますので、事前に準備したアカウントの ID とパスワードを入力して「ログイン」します。



ログインに成功すると「ダッシュボード」が表示されます。ダッシュボード上では制作中の作品を「プロジェクト」という単位で管理します。



(参考) アカウント連携

Google や Microsoft アカウントを利用している場合、アカウント連携を行うことで次回から Monaca Education の ID やパスワードを入力せずにログインできます。

ダッシュボード右上のアイコンから「アカウント設定」を経由して連携ページに進めます。なお、学校の設定によってはアカウント連携は利用できない場合もあります。先生の指示に従って利用して下さい。



(参考) アカウント作成

自身で Monaca Education のアカウント作成を行うこともできます。アカウントを作成する際には先生の指示に従ってください。また、アカウント作成の手順は Monaca Education の公式サイトにて動画付きで解説を行っております。

<https://edu.monaca.io/start>



また、自身でアカウント作成を行う場合はライセンスが適用されないため、有料版の機能を使うためには後からライセンスの設定が必要となります。

プロジェクトを作成する

Monaca Education を体験するために、Monaca Education のテンプレートから「ブロック崩し」を選択して動かしたり改造したりしてみましょう。

プロジェクトを作成する

画面左上の『新しいプロジェクトを作る』ボタンをクリックして下さい。



テンプレート一覧が表示されます。今回はブロック崩しを選択します。



プロジェクトにはプロジェクト名や説明を設定できます。今回はテンプレートの元々のプロジェクト名のまま進めることにします。



ブロック崩しプロジェクトを IDE(統合開発環境) で開く

ダッシュボードで作成したプロジェクトを選択し「クラウド IDE で開く」のボタンをクリックして下さい。画面が切り替わり、プログラムを制作するための IDE(統合開発環境) が展開されます。



各部の名称

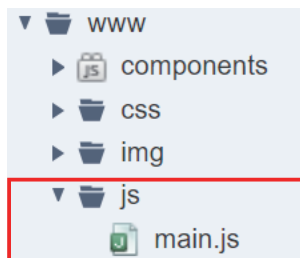
各部の名称は下図の通りです。

メニュー

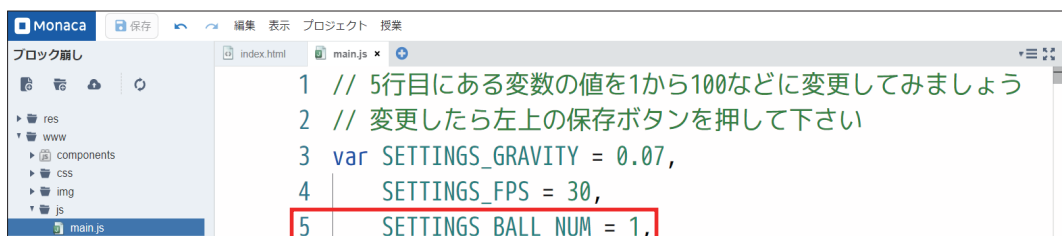


プロジェクトパネルから js/main.js を開いて 5 行目を編集する

フォルダのアイコンの左にある三角形の記号をクリックすると中身が一覧で展開されます。また、ファイルをダブルクリックするとエディタで編集できます。今回は js フォルダを展開して main.js ファイルをエディタで編集します。

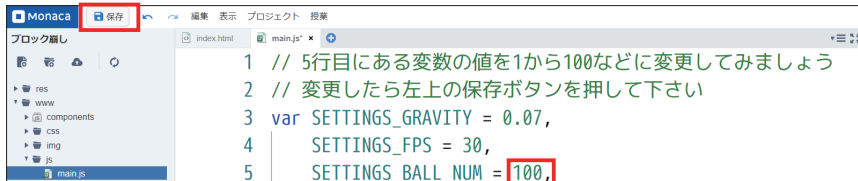


エディタのパネルではプログラムのソースコードを変更できます。今回は main.js の 5 行目を変更して玉の数を 100 個に増やして下さい。



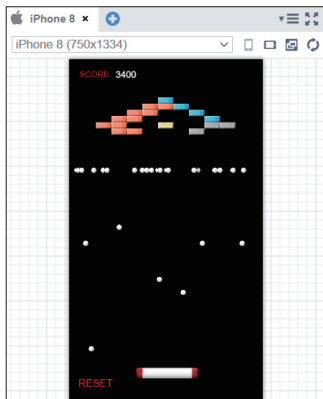
main.js を保存してプレビューパネルに反映させる

変更したら保存も忘れずに行ってください。IDEメニューの左上に保存ボタンがあり、これをクリックすることで保存できます。またショートカットキーでの保存にも対応しており、Ctrl キーを押しながら s キーを押すことでキーボードから手を放すことなく保存できます。



```
1 // 5行目にある変数の値を1から100などに変更してみましょう
2 // 変更したら左上の保存ボタンを押して下さい
3 var SETTINGS_GRAVITY = 0.07,
4   SETTINGS_FPS = 30,
5   SETTINGS_BALL_NUM = 100,
```

保存を行うと自動的にプレビューパネルが更新され、玉の数が増えた状態でブロック崩しが動作します。

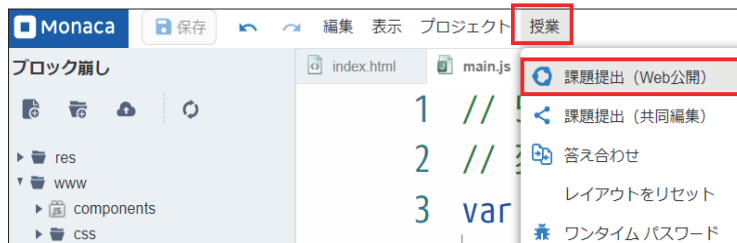


作品を Web に公開する

作品を Web に公開することもできます。公開中の作品はログインなしで自身のスマートフォンなどから動作を確認できます。また、作品の URL を先生に提出することもできます。

Web 公開

メニューの「授業」から「課題提出 (Web 公開)」を選択して下さい。



公開のラジオボタンを「On」にして「適用する」を選択すると作品が公開されます。



QR コードが表示されたら公開中です。



作品の URL を提出するときには「URL をコピー」を使うと便利です。

JavaScript プロジェクトの編集方法

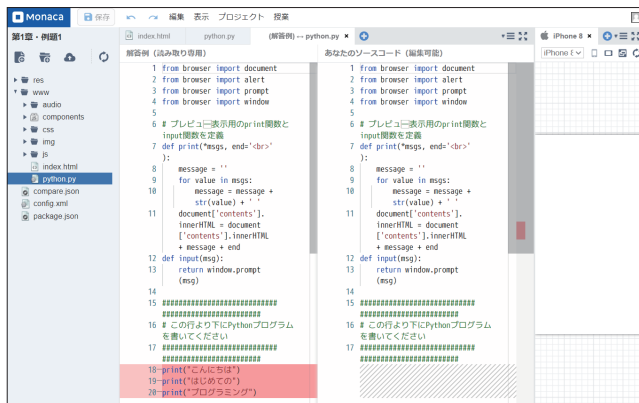
JavaScript プロジェクトのプログラムは index.html ファイルの <script> タグの中に記述して下さい。また、プログレッシブテンプレートを用いて main.js に記述しても構いません。

Python(Brython) プロジェクトの編集方法

Python(Brython) プロジェクトのプログラムは python.py ファイルに記述して下さい。

答え合わせ機能

教材サポートページのプロジェクトは答え合わせ機能に対応しています。例えば「python.py」をエディタで開いた状態でメニュー「授業」の「答え合わせ」を選択すると左に解答例が表示されます。同機能はプロジェクトパネルのファイルを右クリックすることでも呼び出せます。



プロジェクトを開く（セーフモード）

本書の4章以降では「繰り返し（反復）」処理を扱うため、無限ループを発生させてクラウド IDE を開くことができなくなる場合があります。その場合は「セーフモード」で開いて問題の箇所を修正して下さい。セーフモードは「クラウド IDE で開く」ボタンの右にある逆三角のボタンをクリックすることで選択できます。



第1章 順次実行と変数

プログラムを書くとき、コンピュータに計算をさせたり、計算した結果を表示させることができます。コンピュータはプログラムに書いた命令を上から順に実行します。これを**順次実行**といいます。プログラムの中では、計算の結果を一時的に格納しておく**変数**という仕組みを使うことができます。

この章では、短いプログラムを実際に行わせながら、順次実行と変数を学びます。

順次実行

次のプログラムの実行結果を確認しましょう。たった2行ですが、立派なプログラムです。

```
document.write(" こんにちは ");  
document.write(" プログラミング ");
```

プログラムの中では、英字（アルファベット）、日本語（ひらがな・カタカナ）の文字の他、二重引用符（"）や、丸カッコ（（および））などの記号が使われています。

「document.write()」は、（）の中に指定した文字列（文字の並び）を、画面に表示させます。

プログラム言語 JavaScript は初めからたくさんの命令を備えています。「document.write()」はその中の一つです。

```
こんにちはプログラミング
```

このプログラムでは、その document.write() を2回使っています。先に書かれている「document.write(" こんにちは ")」が実行された後に、後ろに書かれている「document.write(" プログラミング ")」が実行されています。

このように、プログラムに書かれている順番の通りに実行する構造を**順次構造**と呼びます。

次のプログラムも実行してみましょう。これも順次構造です。

```
alert(" こんにちは ");  
alert(" プログラミング ");
```

ただし、メッセージは小さなウィンドウ（画面表示をする部品）に表示されます。最初、ウィンドウに「こんにちは」と表示されます。ウィンドウの中のOKボタンを押すと、そのウィンドウが閉じ、すぐに次のウィンドウが表示されます。2個目のウィンドウには「プログラミング」というメッセージが表示されます。

document.write() と、alert() で、メッセージの表示のしかたが違うことを確認してください。

補足 1: JavaScript でプログラムを書くとき、行の終わりには記号「;」(セミコロン) を付けます。

補足 2: document.write() でメッセージを書き出すとき、改行されないのは、HTML を書き出しているからです。HTML では、改行をするためには「
」という HTML タグを付け足す必要があります。

```
document.write(" こんにちは<br>");  
document.write(" プログラミング ");
```

例題 1: 順次実行

次のような実行結果が表示されるように、上のプログラムを変更してください。

```
こんにちははじめてのプログラミング
```

改行も入れてみましょう。

```
こんにちは  
はじめての  
プログラミング
```

補足 1: document.write() や、alert() は、正確には**関数**と呼ばれる部品です。() の中に入れている値を**引数**といいます。関数と引数については、「第 5 章 関数の定義と利用」で詳しく扱います。

補足 2: プログラムの中で英字を使うとき、大文字・小文字は区別されます。たとえば「alert()」と「ALERT()」は別のものとして扱われます。JavaScript には関数 alert はありますが、関数 ALERT は存在しないため、ALERT と書くとプログラムはエラーを発生させます。また、全角文字と半角文字も区別されます。全角の日本語を入力した後に英字を入力するときは、全角・半角に注意してください。

補足 3: プログラムで表示する文字列を扱うときは、二重引用符 (") または一重引用符 (') で囲みます。

変数

次のプログラムを実行してみましょう。

```
let a = "Hello";  
alert(a);
```

実行結果は、「a」ではなく、「Hello」になっています。

```
Hello
```

1行目にある「a」は変数といいます。変数に値を入れて、後で利用することができます。変数に値を入れることを、**代入**するといいます。

最初の行は、JavaScript 言語のキーワード「let」を使い変数 a を宣言し、さらに記号「=」を使って、「Hello」という文字列の値を変数 a に代入しています。

例題 2: 変数に値を代入する

前のプログラムを、次のように編集して、保存します。

```
let a = "Hello";  
a = "こんにちは";  
alert(a);
```

実行結果はどうなるでしょうか？「Hello」と表示されるでしょうか。「こんにちは」と表示されるでしょうか。

補足 1: ここまでのプログラムでは変数の名前として「a」を使いましたが、変数には好きな名前を付けることができます。プログラムや変数の目的・用途に合わせて名前を付けると分かりやすくなります。たとえば、変数の名前を「a」に代えて「aisatsu」とすることが考えられます。変数名に日本語（ひらがな・カタカナ・漢字）を使うこともできますが、プログラムを編集する環境によって読めなくなる可能性があるため、英数字を使うことを薦めます。

補足 2: JavaScript は記号「=」の前後の空白を無視します。前後の空白は、あってもなくてもプログラムの動作は同じです。

数値と計算

ここまで、文字列ばかり扱ってきましたが、数値を扱うこともできます。

数値を計算することもできます。たとえば、記号「+」を使うと、足し算ができます。

```
let b = 10;
alert(b);
let c = b + 10;
alert(c);
alert("b は " + b + " c は " + c);
```

プログラムが少し長くなりました。実行結果とプログラムを1行ずつ見比べて、どんな動作をしたか確認しましょう。

```
10
20
b は 10 c は 20
```

1行目では、変数**b**を宣言して「10」という数値を代入しています。次の2行目で関数**alert()**を使って、変数**b**の値を表示させています。

さらに、3行目では変数**c**を宣言して、値を代入しています。右辺に書かれている、変数**b**と「10」の足し算の結果（20）を代入しています。4行目の「**alert(c)**」で、変数**c**に代入された値を表示しています。

5行目の「**alert()**」では、表示させたい値を記号「+」で連結した文字列を表示させています。

補足：記号「+」のように、計算させるための記号を**算術演算子**といいます。JavaScriptでは文字列に記号「+」を使うと、文字列を結合する（つなぐ）働きをします。

例題 3: 計算結果を表示する

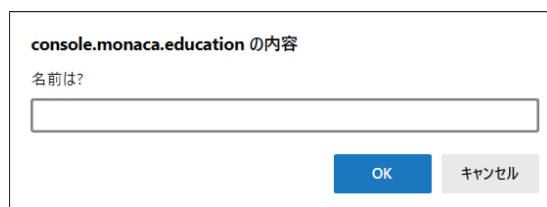
34567 + 123456の結果を表示するようにプログラムを作成してください。その際、値を格納するための2つの変数と、足し算の結果を格納する変数の、合計で3つの変数を使ってください。

入力を受け付ける

次のプログラムを実行してみましょう。

```
let a = prompt(" 名前は?");  
alert(" 名前:" + a);  
let b = prompt(" 年齢は?");  
b = parseInt(b);  
alert(" 年齢:" + b);
```

すると、次のように、入力を促す表示がされます（※この表示は、使用しているブラウザの種類など、動作させている環境によって異なります）。

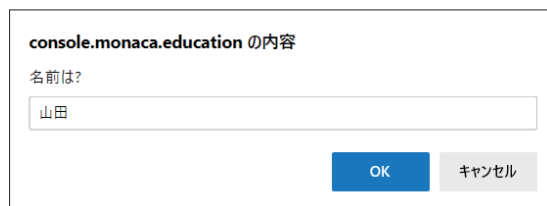


console.monaca.education の内容

名前は?

OK キャンセル

何か好きな値（たとえば、「山田」）を入力し、OK ボタンを押します。

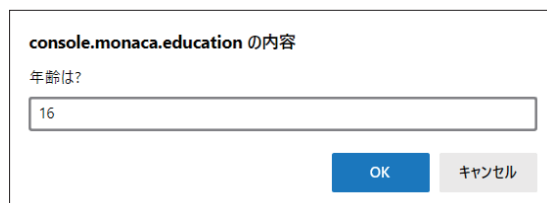


console.monaca.education の内容

名前は?

OK キャンセル

続けて、年齢を入力する画面が表示されるので、数字を入力して、OK ボタンを押します。



console.monaca.education の内容

年齢は?

OK キャンセル

実行結果は次のようになります（※実際に入力した値が表示されます）。

```
名前： 山田  
年齢： 16
```

あらためて、プログラムを確認しましょう。

```
let a = prompt("名前は?");  
alert("名前:" + a);  
let b = prompt("年齢は?");  
b = parseInt(b);  
alert("年齢:" + b);
```

`prompt()` という命令が出てきました。これは、プログラムを動かした人（ユーザー）に、何か値を入力するように求めます。ユーザーが入力した値は、変数に代入して、後の処理で使うことができます。

年齢をたずねている「`prompt()`」が実行されると、ユーザーに入力を求めます。求めに応じてユーザーが値を入力すると、その値は変数 `b` に代入されます。

例題 4: ユーザーの入力を受け取る

次のような処理を順次実行するプログラムを作成してください。

- 1 つめの処理: ユーザーに対して「数字を入力してください」と表示して、入力された値を受け取る。
- 2 つめの処理: 受け取った内容を変数 `a` に代入する。
- 3 つめの処理: 変数 `a` を表示する。

補足資料

表示（出力）と入力

関数名	動作
alert()	<p>カッコの中に入れられた値を、ポップアップする小さなウィンドウに表示する。</p> <p>カッコの中には、値を入れてもよいし ("Hello", 16 など)、変数を入れてもよい (a,b など)。</p> <p>カッコの中で計算をさせてもよい。計算結果がブラウザのポップアップウィンドウに表示される。</p>
document.write()	<p>カッコの中に入れられた値を、画面に表示する。</p> <p>カッコの中には、値を入れてもよいし ("Hello", 16 など)、変数を入れてもよい (a,b など)。</p> <p>カッコの中で計算をさせてもよい。計算結果が表示される。</p>
prompt()	<p>ブラウザのポップアップウィンドウを表示し、ユーザーに値を入力するよう求める。</p> <p>カッコの中に入れた文字列は、入力を求める画面に表示される (" 名前は?", " 年齢は?" など)</p>

値の型を変換する関数

関数名	動作
parseInt()	<p>parseInt("16") のように、整数として扱える文字列が渡されたとき、数値型の値に変換する。</p> <p>a = parseInt("16") とすると、変数 a には 16 という数値の型の値が ("16" という文字列の値ではなく) 代入される。</p> <p>整数として扱えない文字列が渡された場合、プログラムを実行した時にエラーになる。</p>
parseFloat()	<p>parseFloat("174.5") のように、小数点を含む数値として扱える文字列が渡されたとき、数（浮動小数点数）に変換する。</p> <p>浮動小数点数として扱えない文字列が渡された場合、プログラムを実行した時にエラーになる。</p>

算術演算子

演算子	動作
+	数値の足し算を行う。 ※文字列に対して使うと、前後の文字列をつなげることができる。
-	数値の引き算を行う。
*	数値の掛け算を行う。
/	数値の割り算を行う。
%	数値の割り算を行い、余りを得る。剰余計算。 例：10 % 3 -> 1 10 割る 3 は 3 余り 1。% は余りを返す。

問題集

- 問1. `alert()` を3回使って、「おはよう」「こんにちは」「こんばんは」と表示するプログラムを作ってください。
- 問2. 変数 `myouji` に名字を、変数 `nae` に名前を代入した上で、名字と名前を1回で表示するプログラムを作ってください。
- 問3. `prompt()` と `parseInt()`、`alert()` を使い、ユーザーに数字を入力させた後、入力された値に10を足した値を表示するプログラムを作ってください。

本書の教材サポートページは以下の URL からアクセスできます。

<https://edu.monaca.io/template/>

JavaScript で学ぶプログラミング入門

2022 年 4 月 1 日 発行

2023 年 4 月 1 日 初版第 3 刷発行

著者 アシアル株式会社（アシアル情報教育研究所）

協力 株式会社 IMAKE

発行 アシアル株式会社

〒113-0034

東京都文京区湯島 2 丁目 3 1-14

ファーストジェネシスビル

<https://edu.monaca.io/>

(C)ASIAL CORPORATION 2022 Printed in Japan

本書のコピー、スキャン、デジタル化等の無断複製は著作権法上での例外を除き禁じられています。本書を代行業者等の第三者に依頼してスキャンやデジタル化することは、たとえ個人や家庭内の利用でも一切認められておりません。