



アシアル株式会社

Contents 目次

第1章 アプリ開発入門 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
第2章 HTML 入門 画面に文字や画像を表示してみよう	19
第3章 CSS入門 文字に色をつけたり画像のサイズを変えたりしよう	35
第4章 JavaScript入門・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	49
第5章 条件分岐	61
第6章 関数 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	69
第7章 イベント・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	81
第8章 DOM ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	87
第9章 フォーム	99
第10章 いろいろな演算子 様々な計算方法を学んでBMI計算アプリを作成しよう	105
第11章 配列	113
第12章 繰り返し	121

■ はじめに スマホアプリ開発で楽しくプログラミング学習

スマートフォンやタブレットが登場してから10年以上が経ち、ほとんどの人がこれらを日 常的に利用しています。スマートフォンやタブレットは素の状態でも通信機能とウェブサ イトを閲覧するブラウザアプリを搭載していますが、「アプリ」と呼ばれるソフトウェアを 追加インストールすることで、さまざまな機能やサービスを活用できるようになります。 本書では、スマートフォンやタブレットで動作するモバイルアプリを自分で作る方法を解 説します。

アプリを作るというとすごく難しそうな印象を持たれるかもしれませんが、必ずしも高度 な知識や技術が必要になるわけではありません。作りたいアプリの用途や規模にもよりま すが、画面数や機能が限られた簡単なアプリを開発したい場合なら数時間から数十時間の 学習で実現が可能です。

本書ではスマホアプリ開発を通じてプログラミングの基礎を学べます。プログラミングを 行うためのコンピューター言語として「HTML」「CSS」「JavaScript」という3 種類を活用し ます。それぞれの言語の役割は、以下のようになります。

• HTML

文章や画像など、画面に表示する内容を定義します。

画面に表示する内容の色・大きさ・配置といったスタイルを指定します。 · CSS

· JavaScript 「ボタンをクリックしたときに結果を表示する」などのように、アプリに 動きをつけます。

これらの言語はウェブサイトの制作にも利用できる、応用範囲の広い技術です。また特定 の企業やンフトウェアに依存しない技術ですので、一度身に付ければ長く利用することが 期待でき、最初に学ぶ言語としても最適です。



Japi Japi 教材サポートページにアクセスすることで、印刷教材に従って学習する上で役に立つコン

Æ



アプリ開発入門

プログラミングを行うためには、まずプログラム を記述するためのソフトウェアが必要になりま す。そして記述したプログラムをコンピューター にインストールできる形に変換するソフトウェア や、動作確認を行うためのソフトウェアなども必 要です。

こういったプログラミングに必要なソフトウェア を一つ一つ、自分のパソコンにインストールする のは大変です。そこで、プログラミングに必要なソ フトウェアを統合的にまとめた「統合開発環境 (IDE)」を使用します。

■ Monacaとは

Monacaはクラウドで動作する統合開発環境です。インターネット上で利用できるサービス 全般を、雲の上にソフトウェアが置かれているイメージから、「クラウド」や「クラウドサー ビス」と呼びます。開発環境をクラウドに置くことで、自宅と学校のどちらからでもプロ グラミングを行うことが可能となっています。 Monacaは次のような特徴を備えています。

- パソコンに専用のソフトウェアをインストールする必要が無い。
- 少し古めのパソコンでも動作する。
- ・ 開発中のプログラムを先生や友達と共有する機能がある。
- スマートフォンやタブレットで動くモバイルアプリが作れる。
- ・ Webの標準的な技術でアプリ開発できる。

クラウドサービスだからどんなパソコンでも動いて共有も簡単

Monacaは「Google Chromeブラウザ」というWebブラウザから利用できます。このWebブラ ウザがインストールされたパソコンであれば、OS(オペレーティングシステム)の種類や スペック(基本性能)は問いません。作ったアプリの動作確認は普段利用しているスマー トフォンで行うことができます。

また、開発中のプログラムを他の人と共有する機能が搭載されています。作成途中のプロ グラムを先生や友達に見てもらいアドバイスをもらったり、エラーでつまずいてしまった 時に助けてもらったりすることができます。アプリ開発のプロでも、自分以外の人にプロ グラムを見てもらうことで問題がすんなり解決することも多いものです。

モバイルアプリを標準的な技術で開発

スマートフォンで動くアプリを開発するには、さまざまな方法が存在します。Androidや iOSといったスマートフォンのOSごとに別々のプログラミング言語を使わなければなら ない方法もありますが、MonacaではOSの種類を問わずに共通のプログラミング言語(HTML / CSS / JavaScript)を使ってアプリを開発します。Monacaで開発したアプリはAndroid とiOSのどちらでも動作するので、自分や友達、家族などがそれぞれ違う種類のスマート フォンを持っていても、同じようにアプリを動かすことができます。 Monaca の誕生と利用状況

Monacaはモバイルアプリの開発を便利にするために2011年に日本のアシアル株式会社という企業が開発したサービスです。現在、教育向けのMonaca Educationとあわせて40万 人以上の人が利用しており、プロやセミプロだけでなく高校や大学・専門学校の授業でも 幅広く使われています。

Monacaで作られたアプリも増え続けており、既に7万以上のアプリが世に出ています。最 近では有名な企業のアプリでも使われており、代表的なアプリとしてテレビ朝日の映像・ 写真投稿サービス「みんながカメラマン」や、タニタの健康管理アプリ「ヘルスプラネッ ト」、PayPay銀行の「残高確認アプリ」などが存在します。

Monaca Education

日本におけるプログラミング教育の必履修化にあわせて、2015年にスタートしたアシアル の教育事業です。当初は高校・大学・専門学校向けにMonacaの教材とライセンスを提供 するだけでしたが、2019年にはMonaca Education専用サーバーを立ち上げ、教育向けに安 価で高品質なクラウド型のプログラミング環境を日本中に提供しています。また、アシア ル情報教育研究所を設立し、指導者に対する学習機会の提供やコミュニティの運営も行っ ています。

ぷよぷよプログラミング

2020年に日本の代表的なゲーム会社の一つである株式会社セガがリリースしたプログラ ミング教材です。この教材はMonaca Education上で動作する教材として無償で提供されて おり、アカウントがあればすぐに利用できます。興味があればぜひ挑戦してみてください。

Monacaではじめてのプログラミングを書こう

Monacaを利用するためにはまず公式サイトにアクセスしてアカウントを取得します。はじめてのプログラミングでは教育版、ビジネス利用では通常版がお勧めです。



次に右上の「アカウント作成」をクリックして下さい。アカウント作成フォームが表示さ れますので、先生の指示に従ってアカウントを作成します。特に、Google アカウントや Microsoft アカウントと連携してアカウントを作る場合は注意して下さい。連携の場合、 メールアドレスの入力は不要です。

INOTICE Education	アカウント作成		
	メールアドレス 【 必須】		
	バスワード 【 必須】 半角英字と数字を組み合わせた7文字以上 ④		
	アカウント作成ボタンをクリックすると、利用規約に同意 したとみなされます。		~
	アカウント新規作成 or	×	С,
	G Googleアカウントで作成	still 3	
	Microsoftアカワントで作成 国にアカウントをお持ちですか?		

メールアドレスの場合、登録した段階で仮登録状態となり本登録のためのメールが届きます。

Monaca Education	X
ようこそ、Monacaへ!	DX D
Monacaへの登録、ありがとうございます。	·
下記のボタンをクリックして本登録を完了させてください。	
本登録はこちら	
または、URLにアクセスして本登録を完了してください。 https://monaca.education/ja/verify/f7d4d851e7973631037e66b1589fea8b87230ed6	
Copyright © Asial Corporation. All rights reserved.	
メールに記載されたURLにアクセスすることで登録か	_ 「完了します。 □
Contract Education	
本登録完了	
アクティベーションコードをお持ちの方は『ブラン管理』より適応できま す。 10秒後に自動的にダッシュボードに移動します。 移動できない場合 は、以下のリンクより移動してください。	
ダッシュポードに進む	
#2021年12月よりアカウント登録時にはアクティペーションコードやユーザー名を入力しない ように変更となりました。	

Monaca でモバイルアプリを開発しよう

本登録が完了すると10秒後にダッシュボードに移動します。

Monaca Education
新しいプロジェクトを作るインボート
□ ● タグ × オンライン × 作成日順 × 3
新しいプロジェクトを作成してください。 四晩できるプロジェクトがありません。
M/2 CE 07 EF F / M 09 8 E106

この画面では開発中のアプリをプロジェクトという単位で管理します。 画面の左側にプロ ジェクトの一覧が表示されます。

なお無料プランの場合、保持できるプロジェクトは3個までとなっています。4個目のプロ ジェクトを作成するためには古いプロジェクトを消すか有料プランに切り替える必要があ ります。有料プランを利用するためのアクティベーションコードをお持ちの場合は、「プラ ン管理」から適応できます。プラン管理は画面右上のアイコンからアクセスできます。 以上で Monaca のアカウント登録は完了です。

プロジェクトの作成

まずはMonaca の使い方の学習も兼ねて簡単なプログラムを書いてみましょう。 最初に、 プロジェクト(開発中のアプリのこと)を作成します。 [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。



ロジェクトのひな形となるテンプレートを選択する画面が現れます。

テンプレート		
クラシック	プログレッシブ	
クラシック	プログレッシブ	プロック崩し
教育向けの特別な最小限テ ンプレートです。script9 クにLavaScriptを開発迅速 するスタイルを採用してい ます。	main.jsファイルに JavaScriptを記述するスタ イルのテンプレートです。	pixi,jsを用いたブロック崩 しゲームです。難易度やシ ナリオはプレイ後ご自由に 調整してください。
Python	Processing	This is a template for Monaca app.
(Brython)	(p5.js)	
Python(Brython)	Processing(p5.js)	最小限のテンプレート
Brythonを活用してPython を実行できるテンプレート です。python.pyファイルに Pythonを記述して下さい。 実行可能な処理はBrython 公式サイトを参照して下さ い。	p5.isを活用してProcessing を実行できるテンブレート です。sketch,jsファイルに Processingを記述して下さ い。実行可能な処理はp5.is 公式サイトを参照して下さ い。	フレームワークを使用しな い空のテンプレートです。

テンプレートには「サンプルアプリ」として完成した形になっているものから、開発のための土台のみを提供しているものまでさまざまな種類が用意されています。今回は「クラシック」を選択します。

E S HAMILE BOHN

		X
\otimes	新しいプロジェクトを作る	
	 ランブレート 最小限のテンプレート 	
	2 プロジェクトの情報	
	プロジェクト名 はじめてのプログラム	
	説明 	
	作成	

今回はプロジェクト名を「はじめてのプログラム」に変更して「作成」ボタンをクリックします。プロジェクト名は自由につけることができます。あとで見た時にどんなプロジェクトか分かる名前や説明を書くようにしましょう。



「はじめてのプログラム」をクリックし[クラウド IDE で開く」ボタンをクリックします。



画面が切り替わり、MonacaクラウドIDE(以下、IDE)が表示されます。IDEというのは「Integrated Development Environment」の略で日本語では統合開発環境と呼びます。IDEに はプログラミングに必要となるさまざまな機能が用意されています。



上部のパネルが「メニューバー」になります。プログラムを保存する機能やダウンロード する機能などはメニューバーから利用できます。

Monaca
 保存
 ペー 編集 表示 プロジェクト 授業

右側のパネルは「プレビュー」です。プログラムの実行結果が表示されます。「クラシック」テン プレートの場合、なにも表示されません。<body>の部分に「はじめてのプログラム」と記述して、 プレビューに表示してみましょう。メニューバーの左にある[保存ボタン]をクリックすると、 エディタで編集した内容がクラウドに保存され、それによっってプレビュー画面の再読込が行 われ、表示内容が更新されます。

Monaca	🖍 🗠 編集 表示 プロジェクト 授業
はじめてのプログラム	o index.html* ★ 🛟
Image: Cost Cost Cost Cost Cost Cost Cost Cost	<pre>Index.num" * 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4</head></html></pre>
	14 はじめてのプログラム 15
	16

index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html* ×
index.html*

もし、プレビューが自動で更新されない場合はプレビュー画面右上にある円状の矢印ボタ ンをクリックしましょう。

🔹 iPhone 8 × 🕄	
iPhone 8 (750x1334) 🔹 🗋 🖨 🍎	
はじめてのプログラム	to the the shift
小さな一歩ですが、コンピュ	- ターに対して指示を出すことができました。
JavaScript で簡単な命々	令を実行させる
先ほど変更したメッセージは プリを作ることができません のではなく、ユーザーから入 字や画像を後から差し替えた HTMLにJavaScriptというプロ JavaScriptのプログラムはH	HTMLで記述された文章です。HTML単体では「動き」のあるア 。「動き」というのはアニメーションのことだけを指している 力された情報を受け取ったり、ユーザーの操作に合わせて文 りすることなども「動き」といいます。動きのあるアプリは、 コグラミング言語を組み合わせることで作成できます。 TML文章の中にある <script></script>

第一章

Monaca でモバイルアプリを開発しよう

>>> alert()命令によるダイアログ表示

ダイアログとは、画面の前面に表示されるウィンドウのことです。ユーザーにメッセージ を伝えたり、ユーザーからOKまたはキャンセルといった操作を促したりするために使われ ます。 <script>と </script>の間に alert("こんにちは"); という記述を行って保存し、 プレビュー画面で確認してみてください。



Web 公開機能の利用

プレビューはMonacaにログインしていないと確認できないため、自分のスマートフォン でプロジェクトを動かしたり先生に提出したりすることができません。Web公開機能を使 うことで、Web上にプロジェクトを作品として公開し、URLさえ知っていれば誰でもプレ ビューを行えるようにできます。

メニューの[プロジェクト]からWeb公開を選択して下さい。 Monaca 🖥 保存 編集 表示 プロジェクト 授業 ブロック崩し index.html 🗾 main < 共同編集 (function() { 1 Ā Æ \mathbf{O} - 🏠 🚺 エクスポート(zip) 2 3 var SETTINGS_ A エクスポート(Google) 📀 🕨 👿 res 4 SETTINGS 🔻 👿 www 5 SETTINGS_ ✓ エクスポート(Web) is components SETTINGS 6 () Web公開 BETA CSS 7 SETTINGS

選択すると公開の「On / Off」が選択できます。公開をOn にして右下のボタンを押下する ことで作品を公開できます。公開中の作品はログインしていなくても公開用のURL にアク セスすることで閲覧できます。また、QR コードも表示されるため、スマートフォンやタブ レットから読み取って動かすこともできます。

(本機能の紹介はBETA 版の仕様に基づいています、2022 年度以降、一部の機能が変更される可能性があります。最新の仕様はサポートページを参照して下さい。)

公開フォルダ:	www
インデックス:	index.html
公開:	On
https://direct-preview-61cfd5b4e	78885787932d7d2.monaca.educatior
https://direct-preview-61cfd5b4e	78885787932d7d2.monaca.education
https://direct-preview-61cfd5b4e	78885787932d7d2.monaca.education

Monaca for Study(デバッガーアプリ)の利用

先ほど利用した IDE のプレビュー機能や Web 公開機能はウェブブラウザで実行結果を確認 する仕組みになっています。そのため、スマートフォンのカメラやコンパスなどのハード ウェア機能の利用に制限があります。アプリとしてビルドする方法もありますが、ビルド には様々なコストが掛かるため学習目的の場合は「Monaca for Sutdy」という専用アプリ の利用をお勧めしています。本アプリを使うことでプロジェクトを、スマートフォンやタ ブレットのアプリとして動かせます。なお通常版の Monacaでは同様のアプリが「Monacaデ バッガー」の名称で公開されていますので、注意してください。教育版では「Monaca for Study」を利用します。App Store または Google Playで「monaca」というキーワードで検索 し、対応するデバッガーをインストールしてください。なお、本書のレー12章を学ぶだけ であれば Monaca for Studyの利用は必須ではありません。





通常版(Monacaデバッガー)

インストールが終了したらアプリを起動して、先ほどMonacaに登録するときに使ったメー ルアドレスとパスワードを使ってログインしてください。ログインすると開発中のプロ ジェクトー覧が表示されます。動作確認したいプロジェクトの名前をタップすると、実行 結果が表示されます。

なお、アカウント連携機能でMonacaに登録した場合は、パスワードが存在しないため IDE のメニュー[授業]にある「ワンタイムパスワード」機能でログインします。また、IDEの 表示設定を変更することで、Monaca for Studyアプリ内で発生したログやエラー情報を IDE上に表示することもできます。詳しくはサポートページを確認してください。

第2章

HTML 入門

Web ページやモバイルアプリの画面には、さま ざまな色や画像が散りばめられ、とても華やか に装飾されていると思います。しかし実は、画 面の元となっているファイル(「ソース」と呼び ます)には文字だけでページの内容が記述され ています。その記述言語が HTML と呼ばれるもの です。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。

■ HTMLとは

HTML (Hyper Text Markup Language) はマークアップ言語の「つです。マークアップ言語では、 文書が持つ内容をタグと呼ばれる特殊な文字列で囲む形式で記述します。 元々HTMLは、膨大な量の文書を閲覧しやすくする目的で開発されました。例えば、文書の中に 専門用語が出てきた場合、その専門用語について解説されている別の文書をすぐに参照することが出来れば便利です。これを可能にしたのがHTMLによる「リンク」です。HTMLにはリンク以 外にも、文書を構造化したり、画像を参照したりする機能などがあります。

HTML の書き方

HTMLでは、文章やリンク、画像などの画面に表示する内容を「タグ」という文字列で囲みま す。タグとは、画面に表示する内容の種類や役割を表す特殊な文字列です。 タグにはさまざまな種類がありますが、記述方法はどれも同じです。

タグの記述方法と名称 文法

<開始タグ>内容</終了タグ>

タグの記述例 11 例

これは段落です。</p

「開始タグ」と「終了タグ」の部分にはタグの名称が入ります。終了タグの前にはスラッシュ を記述します。開始タグから終了タグまでの全体を「要素」と呼びます。

また、タダの種類によっては終了タグが存在しないものもあります。そのような要素は「空 要素」と呼びます。空要素の場合、スラッシュはつけてもつけなくても構いません。

🔲 文法 空要素の記述方法

<開始タグ>

1 例 空要素の記述例

>

また、各種タグはそれぞれ異なる「属性」を持っています。属性とは、タグにつける付加情報のことで、例えばリンクタグであればリンク先のURLなどを指定します。また、次章で学ぶCSSを一部のタグにだけ適用する場合や、JavaScriptで特定のタグを操作する場合などにもあらかじめ属性をつけておきます。属性は開始タグに記述します。1つのタグに対して複数の種類の属性をつけることが可能です。

📋 文法 属性の記述

<開始タグ 属性1="値" 属性2="値">内容</終了タグ>

19 例 属性の記述例

TOPページへ

属性値はダブルクォート(")のかわりにシングルクォート(')で囲んでも構いません。また、属性の指定順序に決まりはありません。

HTML の構成

HTML文書は、いくつものタグを組み合わせて構成します。記述する際は、要素の中に別の 要素を入れ込んでいく構造(入れ子構造またはネスト構造と呼ぶ)にします。この時注意 しなければならないのは、終了タグの位置です。

1 例 良い例

圓 例 悪い例

<div>これは段落です。</div>

<div>これは段落です。</div>

タグが交差するように配置してはいけません。必ず1つのタグを包むように配置していき ます。 HTML の例

ここでは、第1章で作成したアプリのソースコードを例に解説します。

∞ 解説

🗈 index.html

HTML ①)
<html>2</html>	
<head>3</head>	
<meta charset="utf-8"/>	
<meta <="" content="width=device-width, initial" name="viewport" td=""/> <td></td>	
scale=1, maximum-scale=1, user-scalable=no">	
<meta content="default-src</td><td></td></tr><tr><td>*; style-src * 'unsafe-inline'; script-src * 'unsafe-inline'</td><td></td></tr><tr><td>'unsafe-eval'" http-equiv="Content-Security-Policy"/>	
<script src="components/loader.js"></script> <script src="classic.js"></script>	>
©	
<link href="components/loader.css" rel="stylesheet"/>	
<link href="css/style.css" rel="stylesheet"/>	
<script></td><td></td></tr><tr><td>alert("こんにちは");</td><td></td></tr><tr><td></script>	
<body>8</body>	
はじめてのプログラム	

ここで登場しているのは、アプリを作る上で最低限必要となるタグです。この本の学習範囲内では、基本的に<script>タグや<body>タグの中以外を変更する必要はありませんので、それ以外のタグは削除しないでください。以下に各タグの意味を解説します。

①<!DOCTYPE HTML>

HTMLの最新バージョンで記述された文書であることを表すタグです。終了タグはありません。

(2) <html>

HTML文書であることを表すタグです。文書全体をこのタグで囲みます。

3 <head>

文書全体に関する情報を定義するタグです。このタグ自体はあまり意味を持たず、中に 入っているタグがさまざまな意味を持ちます。

(4) <meta charset="utf-8">

<meta>タグは、メタ情報と呼ばれるHTML文書の補足情報を持つタグです。文書がどの 文字コードで書かれているかといった情報や、スマートフォンなどの小さいサイズのス クリーンで見たときに拡大・縮小する設定などを指定します。

⑤ <script src="components/loader.js"></script><script src="classic.js"></script> <script>タグはJavaScriptのプロダラムを記述するためのタグです。また、 JavaScriptで記述されたファイルを読み込むこともできます。 loader.jsはMonacaでアプリ開発を行う場合に必要なファイルなので、消さないように しましょう。classic.jsはエラーをプレビュー表示するために必要なファイルなので 消しても問題ありませんが、あった方が便利です。

⑥<link rel="stylesheet" href="components/loader.css"> <link rel="stylesheet" href="css/style.css"> <link>タグは外部ファイルを読み込むタグです。ここではCSSで記述されたファイルを 読み込んでいます。loader.cssはMonacaでアプリ開発を行う際に必要なファイルなの ズ 消さないようにしましょう。style.cssは3章以降で解説があります。

⑦<script> alert("こんにちは"); </script>

<script>タグで JavaScript のプログラムを記述しています。

8 <body>

<body>タグは本文を記述するためのタグです。文章や画像などを記述します。

🔓 ポイント 用語説明:ソースコード

ソースは「元となるもの」、コードは「命令文」のことを意味します。ここでは、画面の元と なっている命令文、という意味で使っています。「ソース」と「コード」はそれぞれ単独で 使う場合もあります。

🎍 ポイント 用語説明:文字コード

コンピューターで利用される文字は、内部的には番号で管理されています。文字と番号の 対応関係のことを文字コードといいます。世界各国の文字を表現するためにさまざまな種 類の文字コードが開発されていて、よく使われる文字コードに「UTF-8」「Shift JIS」「EUC-JP」等があります。最新のHTMLでは文字コードにUTF-8を使用することが推奨されていま す。HTML ファイルを保存するときに文字コードの指定を忘れないようにしましょう。

24



アプリの画面を作成する場合は、文章や画像などを表示するために各種要素を<body>要素の 中にタグで記述していきます。HTMLでは非常に多くの要素が用意されていますので、ここでは 1485/11 主要なものを紹介します。

>>>終了タグのある要素

| 要素名 | 概要 |
|--------|---|
| h1 | 見出しを定義します。h1 ~ h6 まであり、h1 が最も高レベル、h6 が最も
低レベルな見出しです。 |
| | 例: <h1> 見出し </h1> |
| | 文章の段落を定義します。 |
| р | 例:
文章の段落を定義します。 |
| | 特に意味を持たないタグです。複数のタグをまとめて扱うときや、四角
い枠を描画したいときに使います。 |
| div | 例: <div>
<h1> 見出し </h1>
段落
</div> |
| а | リンクを定義します。
href 属性・・・リンク先の URL を指定します。 |
| | 例: TOP へ |
| button | ボタンを定義します。 |
| 55000 | 例: <button>ボタン </button> |

>>>空要素(終了タグのないタグ)

| 要素名 | 概要 | |
|--------|---|---|
| img | 画像を参照します。
src 属性・・・画像の参照先を指定します。
alt 属性・・・画像が何らかの理由で表示できなかった場合に、画像の変
わりに表示する文字列を指定します。 | A |
| | 例: | |
| | | |
| >>>すべて | の要素につけられる属性 | |
| 居此夕 | 柳亜 | |

| 属性名 | 概要 |
|-------|--|
| id | 要素を識別するための ID です。文書内で重複する値を指定することはで
きません。 |
| | 例: <div id="header">…</div> |
| class | CSS のクラス名を指定します。(→第3章) |
| | 例: <div class="container">…</div> |

it 。。「頻繁に」 いのの名CALOUCATION 次節では、これらの中でも特に頻繁に使われるリンクと画像の表示についてサンプルを挙



リンクは、ある画面から別の画面へ移動する機能です。リンクを設定する際に重要となるのが、 パスという考え方です。パスとは、HTMLなどのファイルが存在しているコンピューター上の住 所のことです。パスの指定方法には絶対パス指定と相対パス指定の2通りがあります。

📋 文法 リンクの設定

リンク文字列

>>>絶対パス指定

パスを全て記述する方法です。Windowsパソコンの「ドキュメント」フォルダ内に入って いる sample.txtというファイルを表す場合は、「C:¥Users¥ユーザー名¥Documents¥sam ple.txt」がパスになります。また、Webサイトの場合はWebブラウザ上部のアドレスバー に表示される、「https://」から始まる文字列が絶対パスになります。アプリ内にインター ネット上のWebサイトを表示する場合などは、こちらの方法を使います。





≫相対パス指定

現在のファイルから見た、対象ファイルまでの位置を指定する方法です。サイトやアプリ 内の別のHTMLファイルに移動する場合には、こちらの方法を使用します。相対パス指定す る場合、記述方法には以下の決まりがあります。

- ・同一フォルダ内のページに移動する場合はファイル名の指定だけで良い。 ・フォルダとフォルダの区切り、またはフォルダとファイルの区切り文字とし ルダ名の後ろにスラッシュをつける。
- ・一つ上のフォルダは .. という記号で表す。

相対パスの指定方法を以下の図で示します。



実習

今まで記述していたファイルは index.html という名前のファイルですが、このファイル はアプリ起動時に最初に表示されるファイルです。ここから別のHTML ファイルに移動し てみましょう。移動先は [www] フォルダ直下にあらかじめ配置されている、newPage.html です。このファイルをダブルクリックして開くとわかりますが、「新しい画面」という文字 列が <body> タグの中に記述されています

| 第2章_ひな形 | index.html | 💿 newPage.html 🗙 😯 |
|---|---|--|
| 🗘 🔂 📅 🗐 | 1 D<br 2 <ht< th=""><th>OCTYPE HTML>
ml></th></ht<> | OCTYPE HTML>
ml> |
| ► 🐨 res | 3 <he
4</he
 | ad>
<meta charset="utf-8"/> |
| www components | 5 | <meta content="width=device-width, height=device
maximum-scale=1, user-scalable=no" name="viewport"/> |
| 🕨 👕 CSS | 6 | <link href="components/loader.css" rel="stylesheet"/> |
| 🗊 classic.js | 7 | <script src="components/loader.js"></script> |
| index.html | 8 | <script></th></tr><tr><th>💼 monaca.jpg</th><th>9</th><th></script> |
| o newPage.html | 10 <th>ead></th> | ead> |
| compare ison | 11 <bo< th=""><th>dy></th></bo<> | dy> |
| | 12 | 新しい画面 |
| | 13 <th>ody></th> | ody> |
| package.json | 14 <th>tml></th> | tml> |

それでは、index.htmlのタブに戻って15行目に、以下の要素を追記してください。

次の画面へ

</script>

🖹 i r	ndex.html
1	HTML
2	<html></html>
3	<head></head>
4	<meta charset="utf-8"/>
5	<pre><meta content="width=device-width, initial-scale=1,</pre></th></tr><tr><th></th><th><pre>maximum-scale=1, user-scalable=no" name="viewport"/></pre>
6	<pre><meta content="default-src *; style-</pre></th></tr><tr><th></th><th><pre>src * 'unsafe-inline'; script-src * 'unsafe-inline' 'unsafe-eval'" http-equiv="Content-Security-Policy"/></pre>
7	<pre><script src="components/loader.js"></script><script src="classic.js"></script></pre>

11







このように、HTML は画面上に表示したい文字列やリンク、画像などを指定するために利用 します。次章では表示した内容にデザインを適用し、見栄えを良くする方法を学んでいき ます。 \bigcirc

ŧ



第3章

前章で学んだ HTML は、画面に表示する内容を定 義するための技術でした。本章で学ぶ CSS を HTML に組み込んで使うことで、画面を装飾する

前章で作成したプロジェクトを継続して使 用するか、サポートページから本章のひな 形をインポートしてください。

■ CSSとは

CSS (Cascading Style Sheets) は、HTML文書を装飾するための技術です。背景や文字の色 設定を行ったり、文字や画像のサイズ、表示位置を調整したりと、画面にデザインを適用 するために使われます。 色やサイズなどの一つ一つのデザインのことを「スタイル」と呼 びます。「Cascading」は重ね合わせるといった意味がある言葉ですので、CSS (Cascading Style Sheets) はたくさんのスタイルを重ね合わせてデザインを完成させるための文書、 という意味になります。

CSS を HTML ファイルに読み込む方法

CSSは、HTMLファイルの中に読み込む形で利用します。まず、CSSのコードのみをファイル に記述し、拡張子を.cssとして保存します。次に、HTMLファイルに <link> タグを記述し、 href属性に CSS ファイルのパスを指定します。

📋 文法 CSSファイルの読み込み、

k rel="stylesheet" href="CSS ファイルのパス">

なお、本章のひな形プロジェクトやクラシック・テンプレートの index.html には、あらか じめ「style.css」という名前のCSS ファイルを組み込む <link> タグが記述されています。

0	index.htr	ml × 😲 👘 🖓 👘	
	1	IDOCTYPE HTML>	1
	2	<html></html>	
	3	<head></head>	
	4	<meta charset="utf-8"/>	
	5	<pre><meta components="" content="default-src * data: gap: content: https://ssl.</pre></th><th>g</th></tr><tr><th></th><th>7</th><th><pre><script src=" content-security-policy"="" loader.js"="" name="viewport"/></pre>	
	8	<pre>_<link href="components/loader.css" rel="stylesheet"/></pre>	
	9	<pre><link href="css/style.css" rel="stylesheet"/></pre>	
	10	<script></th><th></th></tr><tr><th></th><th>11</th><th></script>	
	12		


CSSを記述する際には「どの要素に対して」「どのようなスタイル」を適用するのか、の2つ の情報が必要です。「どの要素に対して」は「セレクタ」という仕組みでタグなどを指定し ます。また「どのようなスタイル」は「プロパティ」と「値」で指定します。プロパティには 文字の色や背景色またサイズなどさまざまなものが用意されており、適応させたい色や数 値を値として指定できます。

🛅 文法 セレクタとプロパティの記述方法

セレクタ { プロパティ:値; プロパティ:値; と }

セレクタとプロパティの記述例 例

color: red; font-size: 10px;

p {

}

この例では、HTML文書内の タグに対して、文字色を赤に、フォントサイズを IOpx にする、という指定を行っています。

用語説明:px(ピクセル)

コンピューター上に表示される写真や図形などは、ピクセルという点の集合によって描画 されています。点1つが1pxです。一般的にアプリの画面を作るときにはこのpxという単 位を利用します。



セレクタは対象要素を指定する方法で、複数の種類があります。タグ名が同じ要素すべて にスタイルを適用したい場合と、I つの要素に対してのみスタイルを指定したい場合とで は、利用するセレクタが異なります。状況に応じて適切なセレクタを選択しましょう。

・タグセレクタ

対象要素をタグ名で指定します。

・IDセレクタ

対象要素をID属性値で指定します。ID属性値はHTML文書の中で一意となる(重複する値を設定できない)ので、特定の要素 | つだけにスタイルを適用したい場合に利用します。

・クラスセレクタ

CSSにおけるクラスとは、スタイルをひとまとめにして名前をつけたもののことです。 クラスはどのタグに対しても付けることができるので、複数の要素の中から任意の要 素を選択してスタイルを適用したい場合に利用します。

ſ						
	タヤ	レ	ク	タ	の記述方法	Ļ
		~	/	/	VILLEN IL	د

セレクタ	書き方	例	
タグセレクタ	タグ名 {…}	р {…}	
ID セレクタ	#ID {…}	#id1234 {…}	
クラスセレクタ	. クラス名 {…}	.className {…}	
Monaca			

Roh

<u>入</u> 実習		
本章のひな形、もしく cssを開きます。	は前章で作成したプロジェクトを開き、[css]フォルダ内のstyle.	
第3章_ひな形	o index.html o style.css ★ ↔	
	1	
 ▶ ▼ res ▼ www ▶ ≤ components 		
 css style.css classic.js index.html 		
monaca.jpg		
に対してスタイルを適用してみます。以下のユードを記述してください。 サンプルプログラム ■ style.css 1 a { 2 font-size: 30px:		
3 }		
プレビュー画面で確認	すると、リンク文字列の大きさが変更されていることがわかりま	
す。これがタグセレク	'タによる指定です。	

第3章 CSS入門

二 実行結果		
	- BSH	
мопаса		
	(D)X	
	×XO	
続いて、IDセレクタを試してみま index.html 14行目の、「Monaca~	しょう。 リンクを設定している <a> タグに、ID 属性を指定しま	
<i>す</i> .		
₩ サンプルプログラム ■ index. html		
14 <a href="https://edu.mo</th><th>onaca.io/" id="target">Monaca へ 		
次に style.css を開き、先ほど記述した箇所の下に以下のコードを追記します。		
~		





CSSのクラスは、複数のタグに対して同じ属性値を設定することができます。 試しに、<body>内の他のタグにも「class="bright"」という属性を追加してみましょう。属 性を追加したタグの背景色がすべて黄色くなることが確認できると思います。 Moreca Education Hit Hat the multiplication of the state of the state

43

■ プロパティの種類

CSSのプロパティは膨大な種類がありますので、ここでは特によく使われるプロパティを 紹介します。

色を指定するプロパティ

プロパティ	説明	例
color	文字色を設定します。	color: red;
background-color	背景色を設定します。	background-color: red;
border	線の色(および線種と線 の太さ)を設定します。	border: solid 1px red; 線種、線の太さ、線の色の順に 設定します。 solidは直線を表します。

>>>カラーコード

色の表現方法は、「red」や「blue」などの色の名称を指定する方法の他に、カラーコードと 呼ばれる方法があります。コンピューターのディスプレイに表示される色は、光の三原色 (赤、緑、青)を混ぜ合わせて作られています。 それぞれの色の含有量を最小0から最大 255までの数値で表し、16進数にして並べた6桁の数値がカラーコードです。カラーコー ドの先頭には#(シャープ)を付けて記述します。

ドの例

#ff00ff

赤がff(255)、緑が00(0)、青がff(255)なので、原色の赤と青を混ぜた色=紫になります。

カラーコードを調べるときは一般的にPhotoshop、Illustratorなどのデザインやイラスト 制作用のグラフィックソフトを使います。グラフィックソフトが無い場合は、カラーコー ドを算出するWebサービスが多数公開されていますので、各自調べてみましょう。

サイズや位置を指定するプロパティ			
プロパティ 説明		例	
font-size	文字のサイズを設定します。	font-size: l2px;	
text-align 要素内の横方向の配置を設定し ます。		text-align: left; (左寄せ) text-align: right; (右寄せ) text-align: center; (中央揃え) text-align: justify; (均等割付)	
width 要素の横幅を設定します。		width: 100px;	
height 要素の高さを設定します。		height: 300px;	
margin 枠線の外側の余白を設定します。		margin: 20px;	
padding 枠線の内側の余白を設定します。		padding: 10px;	

marginとpaddingはどちらも余白の幅を指定するプロパティですが、余白を取る位置が異なります。以下の図に示すように、marginが枠線の外側、paddingが枠線の内側の余白になります。



margin プロパティと padding プロパティは、ハイフン(-)に続けて方向を表す単語を付け ると、一辺に対してのみ余白を設定することができます。例えば、margin-topは外側の上 余白を設定します。

Mond

方向	外側余白	内側余白	
上	margin-top	padding-top	
下	margin-bottom	padding-bottom	
左	margin-left	padding-left	
右	margin-right	padding-right	

なお、サイズや位置を指定する単位は、px(ピクセル)の他に、%(パーセント)もよく利用されます。%指定の場合は、画面全体または外側にあるタグを100%として計算されます。

実習 プロジェクトのstyle.cssに、以下のコードを追加します。 サンプルプログラム = 🗈 style.css 13 img { width: 30%; border: solid 3px #0000ff; margin: 10px; padding: 20px; 18 } |つ目のwidthプロパティは、要素の横幅を設定するプロパティです。画像の幅を画面全体

に対して 30%に設定していますので、画像が小さく表示されています。

2つ目のborder プロパティは、要素の周囲に線を表示します。カラーコード「#0000ff」は、 青を表しますので、3pxの太さの青い枠線が文字列の周囲に出現します。

残りのmarginプロパティとpaddingプロパティは、余白を設定するプロパティです。枠線の外側に10px、枠線の内側に20pxの余白が作られています。

また結果	the the second with the second
このように、さまざまなCSSプロ にデザインすることができます。	ペティを組み合わせることで、アプリの画面を自由自在

 $-\bigcirc$

Œ



第4章

JavaScript 入門

本章からはいよいよアプリを動かすための技術、 JavaScriptを学んでいきます。JavaScriptは、主にWeb ページやモバイルアプリの画面上の部品を操作する ために利用されます。なおJavaScriptと似たような名 前のプログラミング言語に「Java」というものがあり ますが、これはJavaScriptの略ではありません。Java とJavaScriptは違う言語です。JavaScriptを省略して 呼ぶ場合はJS(ジェイエス)と呼びます。

クラシック・テンプレートを使用するか、 サポートページから本章のひな形をイン ポートしてください。

■ JavaScriptの書き方

第 | 章では以下の JavaScript を実行しました。繰り返しになりますが、JavaScriptはHTML ファイルの <script> タグの中に記述します。



1目例 プログラムA

for(i=0;i<10;i++){alert(i);}</pre>

III例 プログラムB

for(i = 0; i < 10; i++) {
 alert(i);
}</pre>

この2つは、どちらも全く同じことが書かれています。しかし、プログラムAは見づらく、 プログラムBは見やすいと感じられるのではないでしょうか。 このように見やすいコードを書くためのポイントは2つです。

- ・ 単語や記号の間には半角スペースを入れる。
- インデントを正しく設定する。

インデントとは

インデントというのは、文章を記述する際に空白スペースやタブなどを用いて見やすいように字下げを行うことです。プログラムBでは、2行目のalert(i); という命令文が少し 右にずれた位置から開始されています。これがインデントを設定した状態です。

プログラムBではI行目で波かっこが開始されていて、3行目で波かっこが終了していま す。JavaScriptでは波かっこを多用するのですが、プログラム中にかっこがたくさん出て くると、開始かっこに対応する終了かっこが見つけづらくなってきます。

そうでかっこの開始行と終了行を同じ横位置に揃え、かっこの中は右にずらして記述する ことで、かっこの対応関係が一目でわかるようになります。

インデント処理を行うには、キーボードの[Tab]キーを一度押します。Monaca Education の標準設定では半角スペースが4つ分、挿入されます。もちろん、半角スペースを直接入 力しても問題ありません。



コメント

alert命令などの先頭にスラッシュ(/)を2 つ付けると命令は無効化されます。

🛅 文法 一行のコメント

//alert("こんにちは");

JavaScriptでは、// 以降の文字列はコメント(プログラムの実行に影響を与えないメモ 書き)となります。

なお、複数行にわたる文字列をコメントにする場合は、 /* と */ でコメントする範囲を囲みます。

≡ 文法 複数行のコメ

/* コメントとして記述した内容は、 スクリプトには影響しません。 */

コメントには、自分で後からプログラムを読み返したときや、だれか他の人がそのプログ ラムを見たときに、処理内容の理解を助ける説明文を記述します。例えば、「// ラジオボタ ンAが選択された場合は登録処理を行う」といった具合です。

また、今は使わないけれど消去したくはない、残しておきたいコードをコメントにして無 効化する場合もあります。

HTMLのコメントは<!-- -->でしたが、JavaScriptのコメントとは記述方法が異なるので 混同しないように気を付けましょう。



はじめに覚えなければならないのは、JavaScriptで文字列や数値などのデータを扱う方法 です。皆さんが考え事をするとき、頭の中にいろいろな物事(データ)を思い浮かべますね。 それと同じように、コンピューターがプログラムを実行するときには、メモリという装置 上にたくさんのデータを記憶します。メモリ上にデータを記憶するには、まずデータの入 れ物を用意しなければなりません。この入れ物のことを変数と呼びます。



メモリ上に変数を作る作業を、変数の「宣言」と言います。変数に名前を付けて、「この名 前の変数を今から使いますよ」ということをコンピューターに宣言しておくのです。デー タを扱う前には必ず変数の宣言を行います。



var 変数名;

1 例 xという名前の変数を作る

var x;

変数名には自由な名前を付けることができますが、読みやすいプログラムにするためには 何のデータを入れるための変数なのかを推測しやすい名前にしましょう。例えば金額の データを入れる変数であれば、「money」や「price」といった変数名が良いでしょう。 変数の使い方

変数を作った直後は、まだ変数の中には何もデータが入っていない、空っぽの状態になっ ています。変数にデータを入れるには、以下のようにします。

建树脂

😑 文法 変数へ値を入れる

変数名 = 値;

🗊 例 変数×の中に「10」という数値を入れる

x = 10;

1 例 変数xの中に「こんにちは」という文字列を入れる

x = "こんにちは";

※JavaScriptで文字列データを扱う場合は、ダブルクォート(")またはシングルクォート (')で囲みます。

注意しなければならないのは、この「=」記号は算数の「=」記号とは意味合いが異なるということです。算数では左右の値が等しいということを意味しますが、JavaScriptのイコール記号は右辺の値を左辺に入れる、という意味になります。左右が逆になってしまうと正しく動きませんので注意しましょう。

なお、変数へ値を入れることを、値の「代入」といいます。以降も頻出する言葉なので覚え ておきましょう。

宣言と代入は、Ⅰ行にまとめて同時に行うこともできます。



var 変数名 = 値;

JavaScript から画面にデータを出力する

データをアプリの画面に表示するにはさまざまな方法がありますが、ここでは最も簡単な 命令を使って確認してみましょう。

🛅 文法 <body>タグ内にデータを出力する

document.write(表示するデータ);

この命令を使うと、<body>タグ内の一番上の位置にデータを書き込みます。



 \sim

実習

本章のひな形プロジェクトを開き、index.htmlを編集しましょう。

まずは <body> タグ内に「今日も一日がんばりましょう。」という文字列を記述します。

₩ サンプルプログラム ■ index.html	- PSH
15 < <u>body></u> 16 今日も一日がんばりましょう。 17 < <u>/body></u>	
続いて、 <script>タグ内に JavaScript による命令を記述します。</td><td></td></tr><tr><td> </td><td></td></tr><tr><td><pre>10 <script> 11 var today = "2015年09月13日"; 12 document.write(today); 13 </script>	
Monaca Education High	



このままでは <body> タグに文字列を記述したのと変わらないので、次は自動的に現在の 日付が表示されるように変更してみましょう。

■ 今日の日付を取得する

JavaScriptには、さまざまなデータを扱うための便利な命令群があらかじめ用意されてい ます。そして、命令にはたくさんの種類があるので、カテゴリごとに分けられています。 日付を扱うための命令は、「Date」の中に含まれています。

*XO

KIN

日付に関する操作

日付を扱う命令を使えるようにするための準備

var 変数 = new Date();

年を取得する命令

変数.getFullYear();

月を取得する命令

変数.getMonth();

※現在の月から1引いた値が取得される(現在1月なら、0という値が取得される)

日を取得する命令

変数.getDate();

時間を取得する命令

変数.getHours();

分を取得する命令

変数.getMinutes();

秒を取得する命令

変数.getSeconds();

曜日を取得する命令(日曜日~土曜日まで表す、0~6の数値を返す)

THE WILL BE

変数.getDay();

これらの命令を使って、アプリに現在の日付を表示します。

実習

プロジェクトの<script>タグ内を以下のように変更してください。

ルプログラム

🗈 index.html
10 <script> 11 // 日付に関する命令を使えるようにする 12 var date = new Date(); 13 // 年、月、日の取得 14 var year = date.getFullYear(); 15 var month = date.getMonth() + 1; 16 var day = date.getDate(); 17 // 日本の表記にする 18 var today = year + "年" + month + "月" + day + "日"; 19 document.write(today); 20 </script>
完成したら、プレビュー画面で今日の日付が出力されていることを確認して下さい。
少し難しい内容になりましたので、順番に処理を追っていきましょう。
まず、12行目で日付を扱う命令の利用準備を行っています。日付関連の命令を使う場合
は、最初にこの処理が必要となります。
す。この場合の '+」 記号は、昇数と同じて足し昇をするという意味てす。
最後に、18行目で取得した年、月、日を日本表記の日付形式にしています。ここでも「+」記
号が出てきていますが、数値ではなく文字列を「+」記号で繋いでいるので、計算処理はで
きません。文字列を「+」記号で繋いだ場合は、足し算ではなく文字列の連結になります。
▲ ポイント 注意!「+」記号には二通りの意味がある
数値同士を「+」記号で繋いだ場合は足し算、文字列を「+」記号で繋いだ場合は文字列の連 結となる
NOL
これで、今日の日付を表示するアプリは完成です。明日以降、またこのアプリを実行して
みて下さい。日付が更新されていることが確認できるはずです。
このように、JavaScriptを使うことで、いつ見ても同じ画面ではなく、状況に応じて異な
る結果を表示することができるようになります。

第5章

条件分岐

ここまでに実行したプログラムは、上から順番 にすべての処理が実行されていました。しかし これだけでは実用的なアプリは作れません。例 えばゲームではよく、選択肢に応じてその後の シナリオが変化する、といった仕組みがあると 思います。そのようにある条件に応じてその後 行われる処理を振り分けることを条件分岐とい い、アプリ開発には欠かせない考え方です。 本章ではアプリを発展させて、曜日ごとにメッ セージが変わるアプリを作成し、条件分岐を実 現する方法を学びます。

前章で作成したプロジェクトを継続して使 用するか、サポートページから本章のひな 形をインポートしてください。



JavaScriptで条件分岐を実現するには、if文という命令を使います。if文を利用すると、 さまざまな条件によって異なる処理を実行することができるようになります。

THE HALL BENH 🗎 文法 if文の書き方 if (条件式) { 条件式が正しい場合に実行する処理 } else { 条件式が正しくない場合に実行する処理 }

ある条件を指定して、その条件が正しかった場合と正しくなかった場合とで、処理を2方 向に分岐します。else以降は省略が可能なので、正しくなかった場合は何もしない、とい う動作をさせることもできます。 条件式の書き方

if文で条件を指定するための「条件式」は、比較演算子という記号を使って記述します。ある2つの値に対して、「等しい/等しくない/小さい/大きい/以下/以上」のうちいずれかの比較を行います。

比較演算子の種類

演算子	概要	条件式の例	結果	
==	左辺と右辺が等しい場合は正しい	1 == 1	正しい	
!=	左辺と右辺が等しくない場合は正しい	1 != 2	正しい	
<	左辺が右辺より小さい場合は正しい	1 < 1	正しくない	
<=	左辺が右辺以下の場合は正しい	1 <= 1	正しい	
>	左辺が右辺より大きい場合は正しい	1 > 1	正しくない	
>=	左辺が右辺以上の場合は正しい	1 >= 1	正しい	

特によく使うのは、2つの値が等しいかどうかを判定する「==」演算子です。イコール記号 を2つ繋げて記述します。

<u>></u> 実習





getDayが取得する値		
曜日	数値	
日曜日	0	
月曜日	1	
火曜日	2	
水曜日	3	
木曜日	4	
金曜日	5	
土曜日	6	

ソースコード23行目のif文で、今日の曜日を表す数値と、日曜日を表す0が等しいかどう か比較をしています。もし日曜日であれば等しいので、if文の下の「今日は日曜日です」と いうメッセージを表示する処理を実行します。もし日曜日以外であれば、等しくないとい う比較結果になるので、else文の下の「今日は日曜日ではありません」が表示されます。

185419



ここまでではまだ日曜日かどうかの判定しかできていないので、曜日ごとに異なるメッ セージを表示してみましょう。 先ほどの if 文では正しいか、正しくないかの判断を行って E S HALLE BOTH 処理を2方向に分けていましたが、曜日ごとにメッセージを変えるには7方向に分岐をさ せなければなりません。このような場合はelse if文を使います。

i 文法 else if 文

if (条件式1) {
条件式1が正しい場合に実行する処理
} else if(条件式2) {
条件式2が正しい場合に実行する処理
} else {
条件式が正しくない場合に実行する処理
}

else if文は、if文とelse文の間に記述します。else if文は何回でも指定することがで きるので、必要な数だけ条件を指定します。上から順番に条件式1、条件式2…と判定を Monacatoucationth 行っていき、最初に条件式が正しいと判定されたところの処理を実行します。



それでは先ほど書いた23行目からのif文を修正して、一週間分のメッセージを設定しま しょう。波かっこがたくさん出てくるので、開始かっこと終了かっこの数が合っているか どうか気を付けながらタイピングして下さい。





第5章 条件分歧

上 実行結果	
 iPhone 8 × ○ マヨン iPhone 8 (750x1334) マ □ □ ○ ○ 2018年8月23日 木曜日 今日も一日がんばりましょう。 	
	K BOH
今日の曜日に合わせたメ	ッセージが表示されましたね。メッセージの内容は自分の生活に
合わせてカスタマイズし メッセージが変わってい?	てみても良いでしょう。後日、他の曜日にもアプリを起動して、 らことを確認してみてください。
.0	ation
acatolu	
Moli	

Œ



第6章

関数

前章では日付と曜日を表示するアプリを作りま したが、コードを少しずつ追記してきたために だんだんとプログラムが長くなってきました。 それどころか本格的なアプリを作る場合は、コー ドが数百行、数千行、数万行といった途方もな い長さになる場合もあります。そのような長い コードがずらずらと記述されていると非常に読 みづらくなってしまうため、機能単位でコード を切り分けて、「関数」というひとまとまりを作 り、それらを複数組み合わせることでプログラ ムを構築していきます。

クラシック・テンプレートを使用するか、 サポートページから本章のひな形をイン ポートしてください。





ある一連の処理をひとまとめにしたものを関数といいます。機能ごとに処理をまとめるこ とでプログラムが読みやすくなり、何の処理がどこに記述されているかわかりやすくなり ます。また、よく使う処理を関数にしておけば、必要な場面で同じ処理を何度も実行する ことができるようになります。

関数は、処理をするために必要な情報である「引数」を受け取ります。そして、処理が終了 すると「戻り値」と呼ばれる処理結果が返ってきます。



関数の書き方

まずは最も単純な、引数・戻り値ともに省略したパターンから見ていきましょう。

>>>引数も戻り値もない関数

C ----function 関数名() { 処理; }

関数名();

関数は、作っただけでは実行されません。まず関数として名前を付けたものを作っておい て、その関数を呼び出す命令を記述することではじめて関数内の処理が実行されます。つ Monaca Education まり定義と呼び出しの、二段階の手順を踏まなければ関数は動かないということです。



本章のひな形、もしくはクラシック・テンプレートを開き、index.htmlを編集しましょう。<script>タグ内に、以下のプログラムを記述します。


17行目の sayHello 関数を呼び出している処理が走ると、sayHello 関数が定義されている 12行目に飛んで、関数内に記述された処理が実行されます。

なお、関数を定義する処理は記述する順序に関係なく優先的に行われるので、以下のよう に関数の呼び出しが先に記述されていても同じように動作します。





■ 引数がある関数

引数とは、関数が処理をするために必要な情報のことです。例えば、ある数値を2倍する、 という処理を行う関数があった場合、「ある数値」というのがいくつなのかを指定しなけれ ばなりません。この「ある数値」にあたる情報を引数と言います。



関数名(関数に渡す引数1,関数に渡す引数2…);



引数を受け取る関数を作って、実行してみましょう。 先ほど書いたプログラムを、以下のように変更します。「最中 太郎」と記述されていると ころには、自分の名前を書いてみましょう。



■ 戻り値がある関数

戻り値とは、関数の処理結果のことです。ある数値を2倍する関数があった場合、2倍し た値を結果として返却しなければなりません。この結果を戻り値といいます。引数は呼び 机传历制 出したところから関数にデータを渡していましたが、戻り値はその逆で関数から呼び出し たところにデータを渡します。

t

>>>関数の定義(戻り値あり)

function 関数名() { 処理; return 戻り値; }

>>>関数の呼び出し

var 戻り値を入れる変数名 = 関数名();

returnという文が戻り値を呼び出し元に渡す命令です。 呼び出し元は、戻り値を受け取っ たらそれを変数に代入しなければならないので、イコール記号を使って変数へ値を代入す Nonaca Foucatil

OF



引数を受け取る関数を作って、実行してみましょう。 先ほど書いたプログラムの下の19行目以降に、以下のコードを追記します。



まず、26行目のcalc関数を呼び出す処理が実行されて、20行目に飛びます。calc関数内で は、Iから5までの数値を加算した結果を求めて、変数numに代入し、その値を戻り値と して返却しています。するとまた26行目に戻ってきて、戻り値である15という値を変数 resultに代入しています。

Monaca Houcation Hit Hit Hit Market まとめると、戻り値のある関数を呼び出した場合は、以下のような流れでプログラムが動



最後に応用問題として、引数・戻り値の両方があるプログラムを作成してみましょう。 西暦の年を引数として渡すと、和暦に変換して返却する関数を作ります。今回は令和年号 へのみ変換可能な簡易的なプログラムを作成します。



第6章 関数



40行目でconvertYear 関数が呼び出されて、31 ~ 34行目のconvertYear 関数が実行され ます。このとき、引数として西暦の年を渡しています。convertYear 関数内では、受け取っ た西暦から和暦を算出しています。令和元年が2019年なので、西暦から2018を引けば令 和の年に変換することができます。このようにして変換した結果を戻り値として40行目 の呼び出し元に返却しています。

関数は次章以降でも頻繁に利用しますので、使い方は複雑ですがここでしっかりと理解をしておいてください。

第7章

イベント

これまでに作ったアプリは起動時にしか処理が 動いていなかったので、ユーザーは最初に表示 された画面をただ見ることしかできませんでし た。皆さんが普段使っているアプリには、「メ ニューアイコンが押されたらメニューを表示す る」「ボタンが押されたら登録を行う」といった 機能があると思います。このようにユーザーの 操作を受けてから何らかの処理を行うには、「イ ベント」という仕組みを利用します。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。

■ イベントとは

ボタンをクリック/タッチした、画面をスワイプした、などのアプリ上で起こった出来事 のことを「イベント」と呼びます。たとえば、「ユーザーが画面上の部品を押したら、警告 メッセージを表示する」という動作を実現したい場合などに利用します。このイベントと いう仕組みを利用するには、あらかじめ画面上の部品を表すHTML要素に、「このイベント が発生したら、この関数を実行する」といったように要素に対してイベントの関連付けを 行っておきます。

😑 文法 HTML要素にイベントを関連付ける

<タグ名 on イベント名="関数名()">

III 例 ボタンがクリックされたときに test 関数を実行する

<button onclick="test()">ボタン</button×

イベントは、タグの属性として記述します。以下にアプリ制作の際によく使われるイベントをまとめています。パソコンで動くWebサイトであれば 左クリック/右クリック/ダ ブルクリック/マウスカーソルを乗せるといったイベントもあるのですが、モバイルアプ リの場合にはマウス操作をすることがないので、ここでは割愛します。



イベント名	概要
click	HTML 要素がクリック、またはタップされたタイミングで発生します。
load	最初にページが表示されたタイミングで発生します。 このイベントは <body> 要素に対して付与します。</body>
touchstart	モバイル端末の画面に指が触れたタイミングで発生します。
touchmove	モバイル端末の画面に指を触れたまま上下左右に動かしている間、 複数回発生します。
touchend	モバイル端末の画面から指を話したタイミングで発生します。

この中でも、特によく使われるのはclickイベントとloadイベントです。本章ではこの2つのイベントの使い方を学習します。

> 実習

はじめに、clickイベントの使い方について学びます。 本章のひな形プロジェクトを開き、index.htmlを編集しましょう。 まずはボタンが押されたときにメッセージを表示するclickMessage関数を作成します II ~ I3行目に、以下の関数を追加してください。

🗮 サンプルプログラム



つづいて、「クリック/タップされたとき」に発生する click イベントと、 clickMessage 関数を関連付けます。17 行目の <button> タグに、 onclick 属性を追加してください。

🗮 サンプルプログラ

index.html
17 <button onclick="clickMessage()">ボタン </button>

ここまでできたら、保存してプレビュー画面を確認してみましょう。ボタンをクリックす ると、以下のようにダイアログが表示されます。

Nonaca

第7章 イベント

Monaca R存 孫存 第7章_ひな形 で で の の の の の の の の の の の の の の	index.ht	編集 表示 プロジェ ml ★ 😳 HTML	console.monaca.education の内容 ボタンが押されました	ОК	•≡ 23	
▶ platforms	2 3 4	<ntml> <head> <meta_charset;< th=""><th>="""</th><th></th><th></th><th>ポタン</th></meta_charset;<></head></ntml>	="""			ポタン
res	5	<meta content="width=dev</td><td>ice-width, initial-scal</td><td>le=1,</td><td></td></tr><tr><td> is components </td><td>6</td><td><pre>maximum-scale <meta http-eq</pre></td><td>=1, user-scalable=no" name="v</td><td>iewport"/> <mark>uiv=</mark> "Content-Security-Polic	y" content="default-sro	: * data: gap:		
CSS		content: http	s://ssl.gstatic.com; style-	<pre>src * 'unsafe-inline';</pre>	script-src *	
index.html	7	<script src="</td><td>components/loader.js"></scr</td><td>ipt><script <mark>src="class</mark>i</td><td>ic.js"></script>				
.gitignore	8	link rel="st link rel="st	<pre>ylesheet" href="components/ vlesheet" href="css/style c</pre>	loader.css">		
.monacaignore	10	<script></td><td>yresheer mer- ess/styre.e</td><td>22 ×</td><td></td><td></td></tr><tr><td>Config.xml</td><td>11</td><td>function</td><td>clickMessage() { ("ボタンが畑されました")・</td><td></td><td></td><td></td></tr><tr><td>JICENSE</td><td>13</td><td>}</td><td>(319 9 35 (12 0 12))</td><td></td><td></td><td></td></tr><tr><td>package-lock.json</td><td>14</td><td></script>				
package.json	16	<body></body>				
	17 18	<pre></pre>	<mark>ck</mark> ="clickMessage()">ボタン<	/button>		
:のように、以 ⁻ ・イベント発生	下の二単	役階の手順 実行したい	を踏むことで1 関数を作る。	(べ て)を利	用することが	べできます。
このように、以「 I.イベント発生 2.関数とイベン	下の二月 主時に雪 ントを開	没階の手順 実行したい 関連付ける	を踏むことです 関数を作る。	(べてりを利	用することが	べできます。
このように、以 ^て 1.イベント発生 2.関数とイベン 実習	下の二月 主時に ジトを 見	役階の手順 実行したい 関連付ける	夏を踏むことです 「関数を作る。 」。	(べて)を利	用することが	べできます。
このように、以 ^て 1.イベント発生 2.関数とイベン 文字習 次に、ページが	下の二 下 ち に い た 開 い た	役階の手順 実行したい 関連付ける ときに発	iを踏むことです 関数を作る。 。 生する load イ	(べく)を利	用することが	ドできます。 ます。index.html
2のように、以 1.イベント発生 2.関数とイベン を留 なに、ページが 014~16行目	下の二 F F F F F F F F F F F F F F F F F F F	没階の手順 実行したい 関連付ける ときに発 dMessage [を踏むことです 関数を作る。 。 生する load イ 関数を作成して	(ベントを利 ベントの使) ください。	用することが	ドできます。 ます。index.html
2のように、以 ⁻ ・イベント発生 2.関数とイベン 2.関数とイベン 2.関数とイベン 2. 第 3 3 3 3 3 3 3 3 3 4 3 4 3 4 3 4 3 4 3 4 5 4 5 5 5 5 5 5 5 5 5 5 5 5 5	下の二 手 時 に い た に い た の の 、 し の の 二 単 い た に い た の 二 単 い た で 見 い た で 見 い た の た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の し い た の の し い た の の し い た の し い た の の い た の の の の い た の の の い た の の の い た の の の い た の の の の の の し の の の の の の の の の の の の の	没階の手順 実行したい 関連付ける ときに発 dMessage f	iを踏むことです 関数を作る。 。 生する load イ 関数を作成して	(ベントの使) ください。	用することが	ヾできます。 ます。index.html
このように、以 1.イベント発 2.関数とイベ 文 実習 次に、ページが 014~16行目 ゴ 1.100000000000000000000000000000000000	下の二 F の 二 F F E F F E F F E F F E E F E E F E F E F E F E F E F E F E F E F E F E F E F E F E F E F E F E F E F E F E E E E E E E E E E E E E	设階の手順 実行したい 関連付ける ときに発 dMessage	iを踏むことです 関数を作る。 。 生する load イ 関数を作成して	(ベントを利 ベントの使い ください。	用することが	ヾできます。 ます。index.html
2のように、以 1.イベント発生 2.関数とイベン 2.関数とイベン を なに、ページが D 14 ~ 16行目の 14 サンプ A index.html 7 14 fu	下の二月 主時に到 ントを開 開いた こ、loa ルプロ	没階の手順 実行したい 関連付ける ときに発 dMessage コ ク[*]ラム loadMessa	iを踏むことです 関数を作る。 ・ ・ 生する load イ 関数を作成して oge() { ちは ");	(ベントを利 ベントの使い ください。	用することが	ヾできます。 ます。index.html
このように、以 1. イベント発生 2. 関数とイベン 、 関数とイベン 、 実習 次に、ページが つ 14 ~ 16行目の 3. サンプ 1. index.html 14 fu 15 }	下の二月 下 時 に 下 時 に を 開 い た に の の 二 月 い に を 門 に で 開 に で 開 に で に の に の に の の の の の の の の の の の の の	设階の手順 実行したい 関連付ける ときに発 dMessage コクラム loadMessa	iを踏むことです 関数を作る。 ・ 生する load イ 関数を作成して ge() { ちは");	(ベントを利 ベントの使) ください。	用することが	ドできます。 ます。index.html
このように、以て 1. イベント発生 2. 関数とイベン 2. 関数とイベン 2. 関数とイベン 2. 関数とイベン 3. 大いのの 第日 3. 大いのの 3. 大いのの 4. 「u 15. 」 5. 」 5	下の二 手 手 手 時 に 算 い た た 、 loa A イ ベン	设階の手順 実行したい 関連付ける ときに発 dMessage 1 クラム loadMessa c("こんに トと、作成	iを踏むことです 関数を作る。 ・ 生する load イ 関数を作成して gge() { ちは"); にた loadMesso	(ベントを利 ベントの使い ください。	用することか い方を学び: 連付けます。	ドできます。 ます。index.html 19行目の <body></body>



loadイベントはページが開いてすぐに実行されるので、イベントを利用する意味がない ように感じられるかもしれません。今回のようなダイアログを表示するだけの処理であれ ば、イベントを使わずに以下のように記述しても全く同じ結果を得ることができます。



6

では、loadイベントはどのような場合で使うのかというと、次章で学ぶDOMを利用する際 に必要となります。ひとまずここでは、最初に実行されるイベント、と覚えておいてくだ さい。



イベントを扱う方法はタグの属性として記述するやり方の他にも「addEventListener」と いう命令を使うやり方もあります。HTML タグ側を編集せずにイベントを扱えるため必要に Monaca Houcation Hit Hat the multiple of the m 応じて使い分けます。

Ð





DOM

ここまで、JavaScript から画面に文字列を表示す る方法として「document.write」という命令を使っ てきました。しかしこの命令は実際のアプリ開発 ではまず使うことはありません。一般的にアプリ を作るときは HTML で表示内容を定義し、CSS でデ ザインを適用してから JavaScript を記述するた め、「document.write()」で既に完成している画面 の <body> タグ内に文字列を挿入すると、せっかく 綺麗に作った画面が崩れてしまうのです。

そこで、すでに完成している画面に対して変更の 操作を加えるための仕組みとして、DOM という技 術があります。DOM を使うと、表示内容の変更は もちろん、タグの属性やスタイルの変更などがと ても簡単に行えるようになります。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。

■ DOMとは

DOM

DOM (Document Object Model)は、HTMLで書かれた文書の各要素にアクセスするための仕 組みです。アプリを起動して画面が読み込まれたタイミングでDOMという仕組みが働き、 各要素を操作するための機能がJavaScriptで利用できるようになります。

DOM が利用可能になるタイミング

DOMはアプリ起動時に利用可能となるのですが、厳密にはHTMLファイル内のすべてのタ グが読み込まれてからでないと利用することができません。読み込みにかかる時間は | 秒以下なので人間が体感することはまずありませんが、DOMが利用可能になるよりも先に <script>タグ内に記述された JavaScript が実行されてしまうので、JavaScript の書き方 に工夫が必要になります。

という

そこで、前章で学んだ load イベントが役立ちます。このイベントは <body> タグの中身が すべて表示された後に発生するので、load イベント発生時には DOM も利用可能になってい るというわけです。

🎍 ポイント

Monaca

DOMが利用可能になった後に JavaScriptを実行する別の方法として、「プログレッシブ・ テンプレート」を使う方法もあります。このテンプレートは load イベントの記述をしなく ても DOMを操作できます。このテンプレートでは JavaScriptを「main.js ファイル」に記述 する方式を採用しているのですが、ファイルの読み込みが DOM構築後に行われるように調 整されているため1行目から DOMを操作できます。



DOMの機能を使うと、JavaScriptでHTML要素を操作することができます。JavaScriptから HTML要素へアクセスするには、以下の命令を使います。



document.getElementById("ID值")

Elementは要素という意味ですので、getElementByIdという名前はTDを指定してHTML要素を取得する」という意味になります。ここで呼んでいるIDというのは、タグに指定した ID属性値のことです。ですから、この命令を使うためにはあらかじめ対象のHTML要素に ID属性を付けておかなければなりません。

■ 要素の内容を変更する

HTML要素にアクセスすることが出来たら、続けて要素に対する操作を記述します。要素の内容(開始タグと終了タグで囲まれた文字列)を書き換える場合は、以下のように記述します。



上 実行結果		
(iPhone 8 ×) (iPhone 8 (750x1334)		
	-Bolin	
プレビュー画面に	は水色の何もない画面が表示されます。 ただし画面に表示はされてい	
なくても、 <body>タ</body>	グ内にはHTML要素が定義されています。まずは14行目の <p id="mes</td></tr><tr><td>sage">要素の開</p>	始タグと終了タグの間に時間帯に合わせたあいさつを挿入してみます。
まず、 <script></script>		

DOM



>>> && 演算子の使い方

hour >= 5 88 hour <= 10 (時間が5以上)かつ(時間が10以下)

このように複数の条件式を組み合わせることで、より詳細な条件を指定することができるようになります。

そして、実際に画面にあいさつを表示しているのは17行目、19行目、21行目です。それぞれ、「message」という IDを持つ要素の内容に、あいさつの言葉を設定しています。仮に現 在時刻が夜の範囲だった場合は、以下のように 要素が書き換えられています。

>>>実行前

>>>実行後

こんばんは

Monacateducation

93

■ 要素の属性・スタイルを変更する

操作できるのはタグの内容だけではありません。タグの属性も JavaScript で変更するこ とができます。innerHTMLのかわりに属性名を記述し、そこに値を代入すると、既存の属性 値が上書きされます。例えば タグの src属性を変更すれば、表示される画像を切り 替えることができます。

📃 文法 属性の変更

document.getElementById("ID値").属性名 = "属性値";

・ 例 画像を「flower.jpg」に切り替える ・

document.getElementById("target").src = "flower.jpg";

また、以下のように記述することで要素のスタイルを変更することも可能です。

文法 スタイルの変更

document.getElementById("ID値").style.CSSプロパティ名 = "値";

● 一 要素の外側余白を20pxに設定する
 ●

document.getElementById("target").style.margin = "20px";

例外として、「background-color」などのハイフン(-)が含まれるCSSプロパティを変更す る場合は特殊な記述方法になるので注意してください。プロパティ名にハイフンが含まれ る場合は、ハイフンを除去して、ハイフンの後ろの英字を大文字に変更しなければならな いという規則があります。



document.getElementById("target").style.backgroundColor = "blue";



DOM



<body>タグは画面全体を表す特別なタグなので、IDを指定しなくても「document.body」と 指定するだけで要素にアクセスすることができます。この命令が実行されると、CSSファ イルに以下のように記述したのと同じ状態になります。



あらかじめデザインを組み上げた状態のHTMLに対してDOM操作を行うことで、見栄えの良 いアプリを作ることができました。HTMLで表示内容の定義を行い、CSSでデザインの定義 し、JavaScriptで状態に応じて変動する処理を記述する、といったようにそれぞれの役割 に合わせてプログラムを記述することがアプリを作る上でのコツとなります。 Monaca Education Historia Market of Hermiterson

 $-\bigcirc$

Œ

第9章

フォーム

本章では、ユーザーからの情報を入力させる 「フォーム」について学習します。みなさんが普 段使っているアプリにも、ユーザーの情報を入 力する登録画面などがあると思います。フォー ムには、名前やメールアドレスなどの文字列を 入力するテキストボックスや複数の選択肢を表 示するプルダウンメニューなど、さまざまなパー ツが用意されています。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。



checked 属性を付与すると、チェックが入っている状態になります。checked 属性がない場合は、未チェック状態となります。

11 例 チェックボックスの記述例

<input type="checkbox" checked>

ドロップダウンメニュー

複数の選択肢の中からいずれか|件を選択させるためのフォーム部品で

🛅 文法 ドロップダウンメニュー

<select>

<option value="選択肢の値">表示する選択肢</option>
<option value="選択肢の値">表示する選択肢</option>

</select>

選択肢の数だけ、<option>タグを記述します。value 属性に指定した値は、画面上には表示 されませんが、JavaScript でユーザーが選択した値を取得すると、value 属性の値が得ら れます。たとえば、以下の例で選択肢「CCC」がユーザーによって選ばれている場合、Java Script で選択されているものがどれであるかを調べると、「3」という値が取得されます。

/メニューの記述例 例 AAA <select> <option value="1">AAA</option> <option value="2">BBB</option> <option value="3">CCC</option> <option value="4">DDD</option> </select>



	実習
本章の を確認	ひな形プロジェクトを開き、index.htmlの <body>タグ内の内容とプレビュー画面 してください。ユーザー情報の登録画面を模したサンプルとなっています。</body>
∞ €	解説
🖹 inde	ex.html
13 < 14 15 16 17 18 19 	body> <h1> フォームの練習 </h1> <h2> 名前 </h2> <input id="name" type="text"/> <h2> 自動ログイン </h2> <input checked="" id="autoLogin" type="checkbox"/> 次回から自動ログインする <h2> 言語設定 </h2> <select id="languageList"> <option value="japanese"> 日本語 </option> <option value="japanese"> 日本語 </option> <option value="english"> 英語 </option> <option value="chinese"> 中国語 </option> </select>
26	<pre><nu></nu></pre> characteristics of the state of the
27 </th <td>/body></td>	/body>
6行目 メニュ	にテキストボックス、 8行目にチェックボックス、20 ~ 24行目にドロップダウン ーが定義されています。
これら	のフォーム部品に入力された情報をJavaScriptで取得するプログラムを作成しま
す。フォ	オームの入力情報を取得するには、前章で学んだDOMという技術を使います。
<scrip< th=""><td>ht>タグ内に、フォームの入力情報を表示する showProfile 関数を定義しましょう。</td></scrip<>	ht>タグ内に、フォームの入力情報を表示する showProfile 関数を定義しましょう。
A	onaca





第10章

いろいろな演算子

演算子とは、足し算を行う「+」や、比較を行う 「==」などの、各種記号のことです。本章では、 これまでに学んだ演算子以外の、さまざまな演 算子を紹介します。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。

■ 演算子の種類

四則演算子

四則演算子は、足し算や引き算などの算術演算を行うための演算子です。

演算子	概要	条件式の例	結果
+	数値の加算	1 + 1	2
+	文字列の連結	"Hello" + "World"	"HelloWorld"
-	数値の減算	2 - 1	1
*	数値の乗算	2 * 2	4
/	数値の除算	10 / 2	5

※「+」演算子は、計算対象のデータ型に応じて処理内容が変わります。どちらか片方で も文字列であれば、結果は2つの値を連結した文字列となります。

算数の「+」「-」「×」「÷」記号が、JavaScriptではそれぞれ「+」「-」「*」「/」に対応します。 「加算・減算よりも乗算・除算が先に計算される」「丸かっこで括られた範囲を優先的に計 算する」などの計算のルールは、JavaScriptも算数と同じです。

論理演算子

論理演算子は、主にif文の条件式を記述するために使用します。第五章では「今日が何曜日か」を調べるために比較演算子を使いましたが、「第何週目の何曜日か」といった、より 複雑な条件を記述したい場合に論理演算子が必要になります。

演算子	概要	使用例	結果
88	2 つの条件式がどちらも true の場合は true	1 == 1 && 2 == 2	true
	2 つの条件式のどちらかが true の場合は true	1 == 2 2 == 2	true

論理演算子は、2つの条件式を結合します。「&&」演算子はAND演算子とも呼ばれ、2つの条件式がどちらもtrueだった場合に、結果としてtrueを返します。「||」演算子はOR演算子とも呼ばれ、2つの条件式のどちらか一方でもtrueであれば、結果はtrueになります。

「|」はパイプ、またはバーティカルバーと呼ばれる記号です。一般的な日本語キーボードの場合、「¥」記号と同じ位置にキーが配置されています。

複合代入演算子

複合代入演算子は、変数に対して算術演算と代入を同時に行う演算子です。

演算子	概要	使用例	結果(a の値)
+=	左辺の値に右辺の値を加算 したものを代入	a = 1; a += 2;	3
-=	左辺の値から右辺の値を減 算したものを代入	a = 5; a -= 2;	3
*=	左辺の値に右辺の値を乗算 したものを代入	a = 3; a *= 2;	6
/=	左辺の値を右辺の値で除算 したものを代入	a = 10; a /= 2;	5
++	変数に1加算する(インク リメント)	a = 1; a++;	2
	変数から1減算する(デク リメント)	a = 1; a;	0

たとえば、「a += 2」と記述した場合は、「aの現在の値に2を加算して、aを上書きする」と いう意味になります。現在変数に代入されている値を更新したい場合に利用します。 なお、「+=」記号は「+」記号と同様に、数値が演算対象の場合は足し算、文字列が演算対 象の場合は文字列の連結となります。

 \sim

BMI の求め方

四則演算子と論理演算子を使って、BMI計算アプリを作成してみましょう。BMIとは、身長 と体重の相関関係から求められる、体格指数のことです。痩せ・標準体重・肥満などの判 定を行うために用いられる指標です。BMIは、以下の計算式で求めることができます。 C S MAN A SSHIT

BMI の計算式

体重 ÷ (身長²)

※体重の単位はkg、身長の単位はm(cmではないことに注意)

例 身長170 cm、体重60kgの人の場合

 $60 \div (1.7^2) = 20.761...$

BMI 判定表

状態	指標
低体重(痩せ型)	18.5 未満
普通体重	18.5 以上、25 未満
肥満 (1度)	25 以上、30 未満
肥満(2度)	30 以上、35 未満
肥満(3度)	35 以上、40 未満
肥満(4度)	40 以上


本章のひな形プロジェクトを開き、index.htmlの<body>タグ内の内容とプレビュー画面 を確認してください。

∞ 解説

index.html

13	<body></body>
14	<h1>BMI 計算 </h1>
15	<h2> 身長 </h2>
16	<input id="height" type="text" value=""/> cm
17	<h2> 体重 </h2>
18	<input id="weight" type="text" value=""/> kg
19	<hr/>
20	<button onclick="calc()"> 計算 </button>
21	
22	
23	

入力フォームはすでに完成していますので、20行目の「計算」ボタンを押したときに実行 される処理を JavaScript で記述していきます。

RESHR



🔹 iPhone 8 🗙 😋	*≡23		
iPhone 8 (750x1334)	2 ¢		
3MI計算			
白巨			
<u></u> cm			
体重			145
60 kg			1HL
計算			3
		×	
		X	
		X	
		<i>ч.⊈</i>	

まずはじめにユーザーによって入力された身長と体重を取得しています。フォームから 入力される身長はcm単位なので、BMIを求めるにはあらかじめcmからmに変換しておか なければなりません。15行目で身長を100で割ることで単位変換を行っています。 つづいて19行目で計算式に基づいてBMIを求め、23 ~ 35行目でBMI値を元に肥満度の判 定処理を行っています。肥満度は、BMIの数値が指定された範囲内であるかどうかという 条件で判定を行っています。たとえば「肥満(1度)」であるかどうかの判定条件を文章で 正確に表現すると、「BMIが25以上でかつ、BMIが30未満である」といったように、2つの 条件の組み合わせであることがわかります。両方の条件式を満たす必要があるので、この 場合はAND演算子を利用することになります。

また、変数messageには初期値として「あなたは」という文字列が代入されています。そこに、if文の中で複合代入演算子を用いて文字列を追加しています。結果、「あなたは普通体重です」といったように文字列の連結が行われます。

このように、四則演算子や論理演算子を活用すると、より複雑な計算や判定を行うことが できるようになります。たとえば税抜金額を入力して消費税を求めたり、預金額を入力し て利息を求めたりといった、身近なところで役に立つアプリを作成することができます。 Noraca Education Hit Hat the market of the multiplication of the second 今回のサンプルアプリを発展させて、どのようなアプリを作成することができるか考えて



配列

これまで、数値や文字列などのデータは変数と いう入れ物の中に格納していました。扱うデー タの件数が少ない場合は問題ありませんが、10 個、20個、といったたくさんのデータを扱いた い場合、一つ一つ変数を宣言して用意するのは 面倒です。そこで、たくさんのデータをまとめ て格納することができる「配列」という仕組み が用意されています。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。



配列

配列とは、同じ用途で使う複数データを一括で管理するための仕組みです。変数の中に仕 切りがあって、それぞれの部屋の中にデータを格納するイメージです。この部屋のことを 「要素」と呼びます。たとえば100人分のテストの点数をデータとして扱う場合は、100個 分の要素を持つ配列を1つ作成します。

文法 配列の作成 :=

var 配列名 = [要素1, 要素2, 要素3, …];

配列を作成する際は、角かっこの中にカンマ区切りで複数の要素を指定します。要素の数に制限はありません。また、変数と同じように var をつけて宣言することも忘れてはいけません。

1 例 配列の記述例

var fruits = ["りんご", "みかん", "いちご", "ぶどう"];

上記のように記述した場合、要素4件分の配列が作成されます。

≫配列のイメ "みかん" りんご" "いちご" "ぶどう" 0 23 1 fruits

fruitsという大きな変数の中に、4つの要素が含まれているイメージです。1つ1つの要素を識別するために、要素にはインデックスと呼ばれる0から始まる連番が割り振られます。

次に、配列内の各要素を参照し、格納されたデータを取得する方法を解説します。



配列名[インデックス]

🂵 例 配列内の要素を参照する記述例

var fruits = ["りんご", "みかん", "いちご", "ぶどう"]; alert(fruits[2]);

上記のプログラムを実行すると、インデックスが2番目の要素に入っている値、「いちご」が 表示されます。インデックスの先頭は | ではなく0から始まることに注意してください。

配列を利用することで、冗長なプログラムをすっきりと記述することができるようになり ます。第5章でif文を使って今日の曜日を判定するプログラムを作成しましたが、あのプ ログラムは実はif文を使わなくとも配列だけで作成することができます。if文は複雑な 条件分岐を行う際に有効な文法であって、今日の曜日を判定するだけの単純なプログラム であれば、配列を使ったほうが適切であるといえます。

🚥 解説 if文による曜日判定

Ē]i	index.html	
22	Vá	ar weekday = date.getDay();
23	it	f(weekday == 0) {
24		document.write("日曜日");
25	}	else if(weekday == 1) {
26		document.write("月曜日");
27	}	else if(weekday == 2) {
28		document.write("火曜日");
29	}	else if(weekday == 3) {
30		document.write("水曜日");
31	}	else if(weekday == 4) {
32		document.write("木曜日");
33	}	else if(weekday == 5) {
34		document.write("金曜日");
35	}	else if(weekday == 6) {
36		document.write(" 土曜日 ");
37	}	else {
38		document.write("エラ <u>ー!");</u>
39	}	

∞ 解診 ■ index.h	む 配列による曜日判定 tml
22 23 曜日 ' 24	var weekday = date.getDay(); var list = ["日曜日", "月曜日", "火曜日", "水曜日", "木曜日", "金 ', "土曜日"]; document.write(list[weekday]);
最後に、配	列の要素数を取得する方法を解説します。本章の実習では利用しませんが、次
章の繰り返	して配列の要素数の取得が必要となります。
直 文法	5 配列の要素数を取得する
配列名.le	ength
1 例	配列の要素数を取得する記述例
var fruit alert(fru	s = ["りんご", "みかん", "いちご", "ぶどう"]; its.length);
上記のプロ	ログラムを実行すると、要素数の「4」が表示されます。
•	

配列

第||章



🗈 index.html

13	<body></body>
14	<h1> 心理テスト </h1>
15	あなたが森の中を一人で散歩していると、ある生き物に出会いました。その生
	き物はどれですか?
16	<select id="choiceList"></select>
17	<option value="0">鳥</option>
18	<option value="1"> リス </option>

<option value="2"> クマ </option> 19 20 <option value="3">人間 </option> 21 <option value="4">妖精 </option> 22 </select> 23 <button onclick="showAnswer()">回答する </button> 24 <hr>> 25 26 </body>

ユーザーに心理テストの回答をプルダウンメニューの中から選択させ、それに対応した結 果を表示するプログラムを作成します。16 ~ 22 行目で定義しているプルダウンメニュー が、心理テストの選択肢になります。ユーザーが選んだ選択肢に応じて、<option>タグの value 属性値である0~4のいずれかの値が JavaScript に渡されることになります。



第||章

配列

上 実行結果	
IPhone 8 × Image: Second state IPhone 8 (750x1334) Image: Second state	
心理テスト	
あなたが森の中を一人で散歩していると、ある生 き物に出会いました。その生き物はどれですか?	
この心理テストは、あなたが一番大事にしている ものを表しています。 あなたが一番大事にしているものは、「友情」で す。	

11~17行目では、あらかじめ心理テストの結果を配列 answerList として定義しています。 先頭から順番にインデックスが割り振られますので、「友情」が0番、「愛」が1番、最後の 「夢」は4番になります。

19行目の showAnswer 関数は、「回答する」ボタンを押したときに実行される関数です。こ の関数の中で、心理テストの結果をの開始タグと終了タグの間に挿 入しています。

2l 行目では、🐘 (ダブルクォート2つ)をp要素の innerHTML に代入しています。 ダブル クォート 2つは空文字といい、その名のとおり空っぽの文字列であることを表していま す。「回答する」ボタンを二回押したときに、一回目に表示された結果メッセージを消して、 空っぽの初期状態に戻したいので、この処理が必要になります。

最後に22、23行目で結果メッセージを挿入しています。answerList[choiceNo]が、ユーザー が選んだ回答に対する結果になります。仮にユーザーが「鳥」という回答を選択した場合、 choiceNoには<option>タグのvalue属性値である「0」が代入されています。 つまり answer List [0] となり、配列の0番目の要素である「友情」が表示されるという仕組みです。

配列を使うと扱うことのできるデータの幅が増えることがわかったかと思います。大量の情 報を扱う場合は、変数を個別に用意するのではなく配列を利用すると覚えておきましょう。 また、配列を扱う際は、次章で学ぶ for 文と組み合わせる方法がよく用いられます。 配列 . *** と for 文を組み合わせると、要素 | つ | つに対して順番に処理を実行することができるよ

第11章	配列
------	----

Monaca Education Hit HAM OF A CONSTRUCTION OF A CONSTRUCT OF A CON

 $-\bigcirc$

Œ



繰り返し

プログラムの作成中に、同じ処理を何回も繰り 返し実行したい、という場面が出てくるはずで す。例えば I ~100 までの数の合計を求める処 理や、100個分の要素を持つ配列のデータを I 件 ずつ表示する処理などが挙げられるでしょう。 同じ処理を何度も実行したい場合、コピー&ペー ストで処理を増やす方法ではキリがないので、 繰り返しを行うための特別な文法を使って記述 します。

実習の前にサポートページから本章のため に用意されたひな形をインポートしてくだ さい。





ある処理を繰り返し実行するには、for文という命令を利用します。for文は、処理を繰り 返し実行する中で、1回繰り返すたびに変数の値が移り変わって行くという特徴がありま す。この値が移り変わって行く変数のことを、「カウンタ変数」と呼びます。 * BSHA

Ⅲ 文法 for 文

for (初期化式; 継続条件式; 増減式) { 繰り返す処理 }

for 文の記述方法は少々複雑です。丸かっこの中に、セミコロン(;)で区切って3つの式を 記述します。それぞれの式の意味は以下の通りです。

式の種類	概要	実行するタイミング
初期化式	カウンタ変数を初期化する代入式	最初の丨回だけ実行される
継続条件式	繰り返しを継続する条件式	回分の繰り返し処理の最初に毎回実 行される
増減式	カウンタ変数の値を増減させる計算式	Ⅰ回分の繰り返し処理の最後に毎回実 行される

少し難しい内容になりますので、実習を通して覚えていきましょう。

🍾 実習

「繰り返しの練習」プロジェクトを開き、index.htmlに以下の処理を追加してください。

ンプルプログラム ≣ 🖹 index.html 10 <script> for (var i = 1; i <= 10; i++) {</pre> document.write(i); } 14 </script>



カウンタ変数の名前は通常の変数と同様に自由ですが、慣例としてiという名前がよく使われます。

for 文は配列を扱う際にも重宝します。配列のインデックスをカウンタ変数で指定することで、配列の各要素に対して | 件ずつ順に処理をすることができます。

🔪 実習

index.htmlの15行目以降に、ボタンを押したときに実行されるinsert関数を追記してください。

₩ サンプルプログラム

🖹 i r	ndex.html
10	<script></th></tr><tr><th>11</th><th>for (var i = 1; i <= 10; i++) {</th></tr><tr><th>12</th><th><pre>document.write(i);</pre></th></tr><tr><th>13</th><th>}</th></tr><tr><th>14</th><th></th></tr><tr><th>15</th><th><pre>function insert() {</pre></th></tr><tr><th>16</th><th>var images = [</th></tr><tr><th>17</th><th>"strawberry.png",</th></tr><tr><th>18</th><th>"orange.png",</th></tr><tr><th>19</th><th>"grape.png",</th></tr><tr><th>20</th><th>"peach.png",</th></tr><tr><th>21</th><th>"melon.png"</th></tr><tr><th>22</th><th>];</th></tr><tr><th>23</th><th></th></tr><tr><th>24</th><th>for (var i = 0; i < images.length; i++) {</th></tr><tr><th>25</th><th>document.getElementById("imageList").innerHTML += "<img src=" +</th></tr><tr><th></th><th>images[i] + ">";</th></tr><tr><th>26</th><th>}</th></tr><tr><th>27</th><th>}</th></tr><tr><th>28</th><th></script>



ボタンをクリックすると、画像が5つ表示されます。



24~26行目で、for文を使って<div id="imageList"></div>の開始タグと終了タグの中 にタグを挿入する処理を5回繰り返しています。

for 文の3つの式を順に見ていくと、iは0から始まり、「images.length」未満の間繰り返し、 | つずつ増えていく、と指定されています。images.lengthは、配列 imagesの要素数です。5 つの画像ファイルを要素として持っているので、「5」を表します。つまり、iの値は0,1,2,3,4 というように変化していきます。0,1,2,3,4という値は、配列 images のインデックスと等し いので、ちょうどカウンタ変数iの値が配列のインデックスと対応することになります。 走的建树儿也是的 この for 文が実行されると、以下のように HTML タグを記載したのと同じ状態になります。

<div id="imageList">

 </div>

繰り返しを使えば、数値でも文字列でも画像でも、大量のデータを扱う処理をほんの数行 Monaca Education High の記述で済ませることができます。

Moraca Education Hit Hat the but the state of the state o

 $-\bigcirc$

 \bigcirc

 \oplus

本書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/

KA Tipt Monaca で学ぶアプリ制作入門 ~ HTMLxCSSxJavaScript 編~

2022年4月1日初版第1刷発行

MonacaFducation 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31-14 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。