
Monaca で学ぶ
チャットアプリ開発

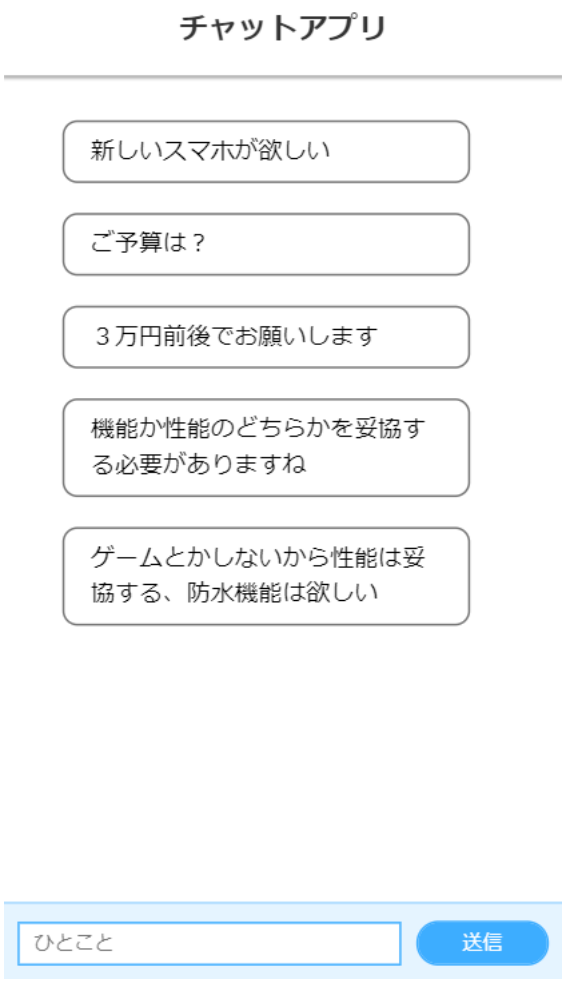
序章：チャットアプリの仕様と開発計画

アプリの種類と難易度

- **基本版**
  - Lv1:一言チャット
  - Lv2:一言チャット（order&limit）
- **カスタム版 (easy)**
  - Lv3:名前付きチャット
  - Lv4:吹き出しチャット
- **もっとカスタム版 (hard)**
  - Lv5:アイコンチャット
  - Lv6:真アイコンチャット
- **最終カスタム版 (hard)**
  - Lv7:起動画面付きチャット

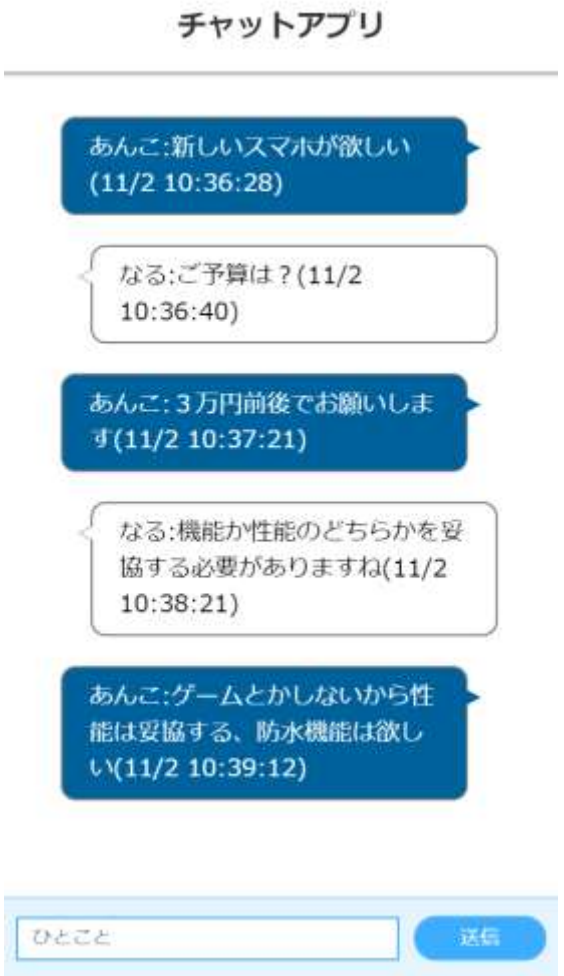
基本版の機能

- メッセージを投稿できる
- メッセージを一覧表示できる



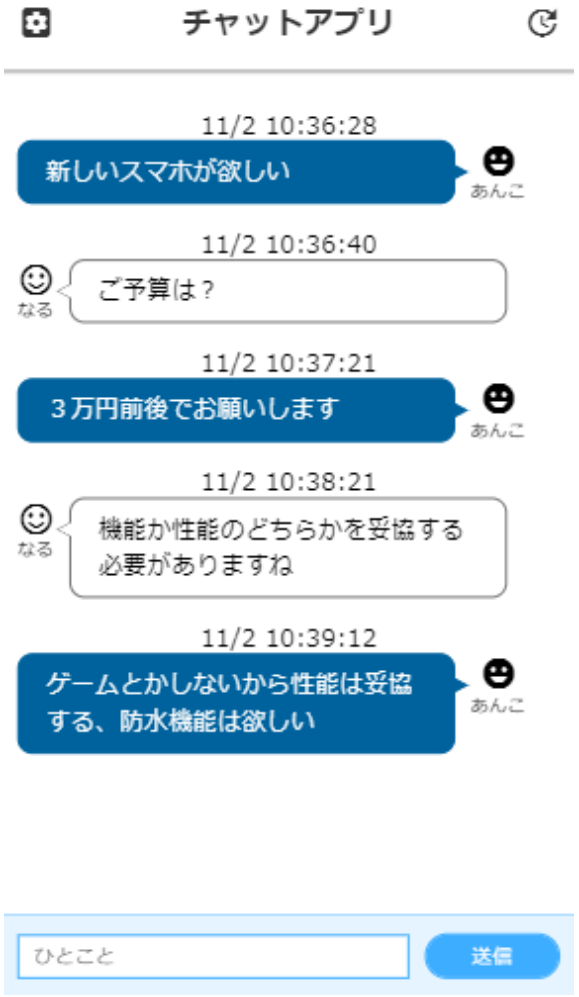
カスタム版の機能

- 名前とメッセージを投稿できる
- 投稿者の名前と投稿日時が表示される
- 自分の投稿は吹き出しの
  向きなどが変わる



もっとカスタム版の機能

- 最新の投稿一覧を取得できる（リロード）
- 投稿時にアイコンも登録できる
- 投稿一覧でアイコンが表示される
- 名前と投稿日時がメッセージの外に表示される



最終カスタム版

- 開始画面が新設される
- アイコンを一覧から選べる
- チャットルームが選べる

チャットアプリのデータ設計

- chat
  - name：投稿者
  - message：投稿内容
  - icon：アイコンの名前
  - createDate：投稿日時 ※DBが自動的に記録します

実データの例

objectId	icon	message	name	createDate
X6UWjaKSkj9dgpEb	emoji_emotions	ゲームとかしないから性能は妥協する、防水機能…	あんこ	2019-11-05T10:39:12.612+09:00
Ppq232SpxusQQBxP	sentiment_satisfied_alt	機能か性能のどちらかを妥協する必要がありますね	なる	2019-11-05T10:38:21.208+09:00
Dx5P6FuB98g3QSOS	emoji_emotions	3万円前後でお願いします	あんこ	2019-11-05T10:37:21.220+09:00
TKEmPh6ad6OkPFKT	sentiment_satisfied_alt	ご予算は？	なる	2019-11-05T10:36:40.493+09:00
HFamSta6CY93H7Hj	emoji_emotions	新しいスマホが欲しい	あんこ	2019-11-05T10:36:28.418+09:00

主要な独自関数

- read()
  - DBの投稿内容を取得して一覧表示する。
    Lvが上がるにつれて表示が複雑になるため、read()の処理もどんどん増えていく。
- send()
  - DBにメッセージを登録する。
    Lvが上がると名前やアイコンも登録するが、処理はあまり増えない。
- init()
  - アプリの初期化処理を行う。
    基本版ではread()関数を実行するだけだが、
    カスタム版では名前などの初期設定に活用されている。
- start()
  - 最終カスタム版のみ使用。
    スタート画面のフォームに記述された設定を読み込み、アプリをスタートさせる。

主要な標準関数

- new Date()
  - 日付オブジェクトを作成する。
  - 引数無しの場合は現在時刻のオブジェクトになる。
  - 引数で日付を指定した場合、指定した日付のオブジェクトになる。
- Dateオブジェクト
  - .getMonth() メソッド：月を返す。実際には+1する必要がある。
  - .getDay()メソッド：日を返す。
  - .getHours()メソッド：時を返す。
  - .getMinutes()メソッド：分を返す。
  - .getSeconds()メソッド：秒を返す。

主要な標準関数

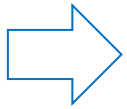
- document.getElementById()
  - DOMオブジェクトをID名で探して取得する。
  - 投稿一覧を書きかえたり、メッセージフォームの値を取得したりする時に使います。
- document.createElement()
  - DOMオブジェクトを新規作成する。
  - 投稿一覧に追記するDOMを作成するときに使います。
  - LV5以上は投稿内容が複雑になるため、この関数を多用します。

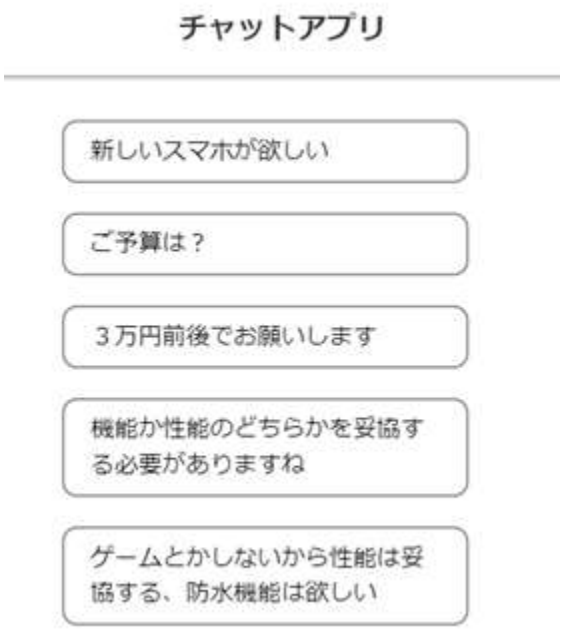
DOMオブジェクトのプロパティ確認

- .valueプロパティ
  - フォームの値を取得する。
- .innerHTMLプロパティ
  - 要素の内容をHTMLレベルで取得ないし変更する。
- . textContentプロパティ
  - 要素の内容をTEXTレベルで取得ないし変更する。
- .appendChild()メソッド
  - DOM要素を子要素として末尾に追加する。

リストタグとチャット表示の対応（基本版）

```
<ul id="chat">
  <li class="box">新しいスマホが欲しい
</li>
  <li class="box">ご予算は？</li>
  <li class="box">３万円前後でお願いしま
す</li>
  <li class="box">機能か性能のどちらかを
妥協する必要がありますね</li>
  <li class="box">ゲームとかしないから性
能は妥協する、防水機能は欲しい</li>
</ul>
```





チャットアプリ開発

チャットアプリ開発

Lv 1

プロジェクトのインポート

初期状態（HTML）

Lv1ではHTMLの編集はありません

```
<body onload="init()">
    <header>
        <h1>チャットアプリ</h1>
    </header>
    <section id="main">
        <ul id="chat">
        </ul>
    </section>
    <footer>
        <form id="message_form">
            <input type="text" id="message" value="" placeholder="ひとこと">
            <input type="button" value="送信" id="sendbtn" onclick="send()">
        </form>
    </footer>
</body>
```

初期状態（HTML）

header

h1

section id="main"

ul id="chat"

footer

form id="message_form"

初期状態（JavaScript）

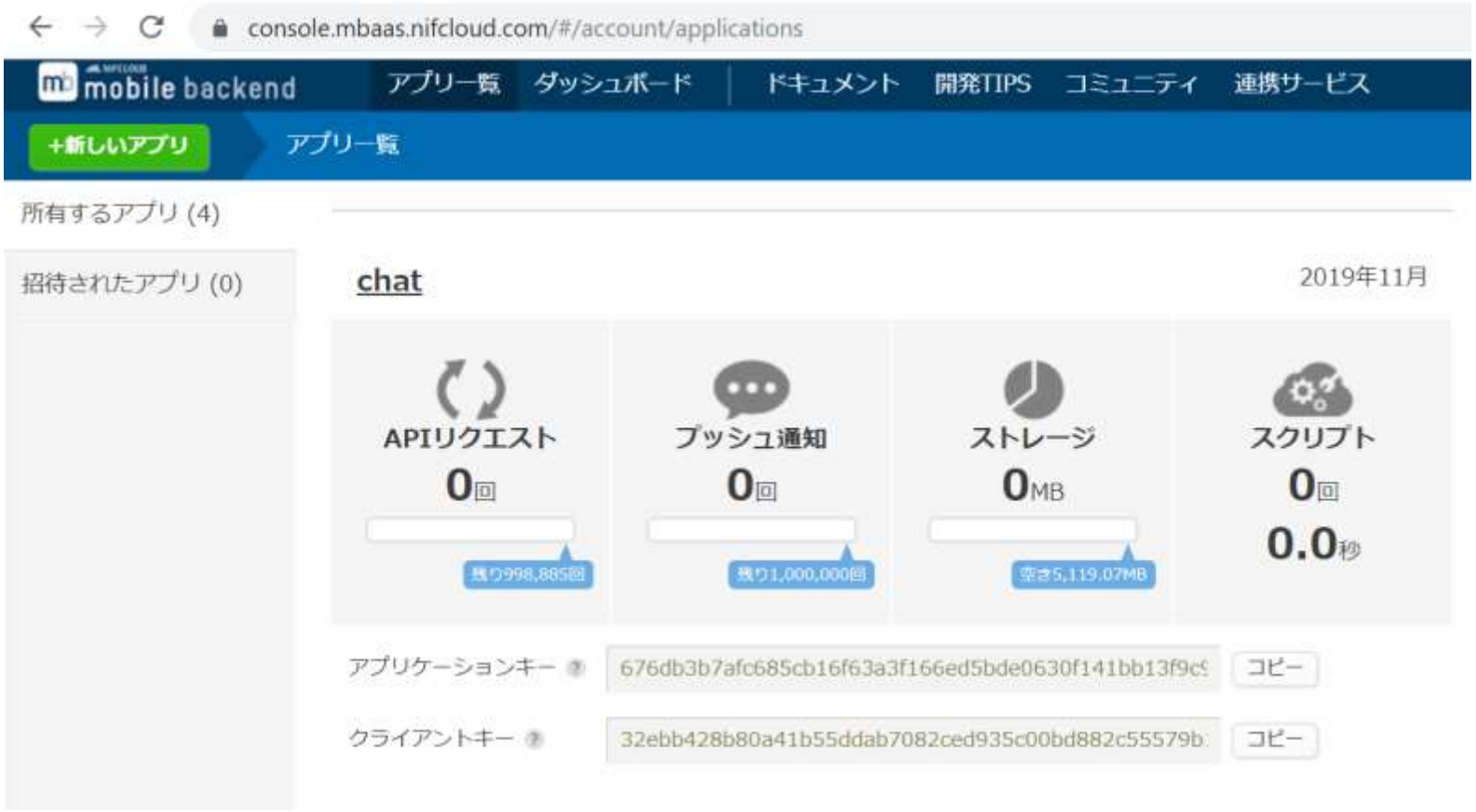
Lv1で実装すること

- 値の設定
  - DB接続情報
- 関数
  - read()
  - send()

```
// アプリの設定
var room = "chat";

// ncmbのデータベース(DB)に接続
var APIKEY = "*****";
var CLIENTKEY = "*****";
var ncmb = new NCMB(APIKEY, CLIENTKEY);

// チャットテーブルのオブジェクトを作成
var chatTable = ncmb.DataStore(room);

// アプリの初期化処理
function init(){
    read();
}

// DBからチャットデータを取得して描画する
function read() {
}

// 新規投稿をDBに登録する
function send() {
}
```

ncmbの接続情報取得 (アプリ新規作成の場合)

ncmbの接続情報取得（既にアプリがある場合）



DB接続情報設定（JavaScript）

- キーの情報をncmbからコピーして入力してください

```
// アプリの設定
var room = "chat";

// ncmbのデータベース(DB)に接続
var APIKEY = "*****";
var CLIENTKEY = "*****";
```

read()関数の実装（JavaScript）

```
function read() {
    // 画面の初期化
    document.getElementById("chat").innerHTML = null;
    // チャットデータの取得と反映
    chatTable.fetchAll().then(function(records){
        for (var i = 0; i < records.length; i++) {
            // 投稿を表示するためのリストタグを生成
            var dom = document.createElement("li");
            // リストタグにClass属性を付与
            dom.classList.add('box');
            // 投稿内容を作成
            dom.textContent = records[i].get("message");
            // 投稿内容をチャットエリアに追記
            document.getElementById("chat").appendChild(dom);
        }
    })
    .catch(function(error){
        // もしfetchAll()が失敗したらエラーをログに書き出す
        console.log(error);
    });
}
```

チャット初期化

DB取得して
件数分繰り返し処理

データを元にliのDOM生成

チャットに追記

send()関数の実装（JavaScript）

```
function send() {
    // フォームの値を取得
    var message = document.getElementById("message").value;
    // チャットレコードの作成と反映（変数に入れることを省いている）
    new chatTable()
        .set("message",message)
        .save()
        .then(function() {
            read();
        })
        .catch(function(error){
            // もしsave()が失敗したらエラーをログに書き出す
            console.log(error);
    });
}
```

フォームから
メッセージを取得

DBにメッセージを保存

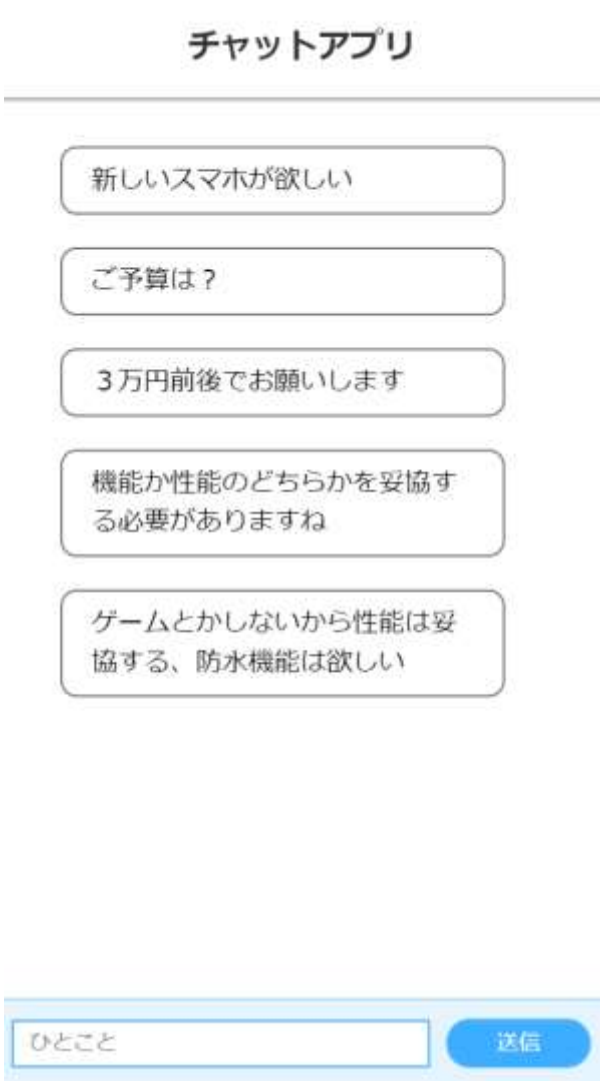
read()関数を呼んで
チャット表示を更新

チャットアプリ開発

Lv2

Lv2で実装すること

- 取得件数の制限
  - limitを掛ける
- 取得順の変更
  - 新着順
- 表示順の変更
  - 古い順



limitの宣言

```
// アプリの設定
var limit = 5;
var room = "chat";
```

limitを宣言、今回は5件

取得プログラムの変更

read()関数内

```
// チャットデータの取得と反映
chatTable
.order("createDate",true)
.limit(limit)
.fetchAll()
.then(function(records){
    records.reverse();
    for (var i = 0; i < records.length; i++) {
```

DB取得時に作成日でorderを掛ける

DB取得時にlimitを設定

取得結果を逆順ソート

カスタム版(easy)

チャットアプリ開発

Lv3

Lv3で実装すること

- 名前の投稿
  - フォームを増やす
  - send()関数を改造
- 名前の表示
  - read()関数を改造
- 日付の表示
  - read()関数を改造

HTMLの変更

```
<form id="message_form">
    <input type="text" id="myName" value="" placeholder="なまえ">
    <input type="text" id="message" value="" placeholder="ひとこと">
    <input type="button" value="送信" id="sendbtn" onclick="send()">
</form>
```

名前の欄を増やす

send()プログラムの変更

```
// フォームの値を取得
var message = document.getElementById("message").value;
var myName = document.getElementById("myName").value;
```

名前の取得を増やす

send()プログラムの変更

```
function send() {
    // フォームの値を取得
    var message = document.getElementById("message").value;
    document.getElementById("message").value = "";
    // チャットレコードの作成と反映
    new chatTable()
        .set("message",message)
        .set("name",myName)
        .save()
        .then(function() {
            read();
        })
        .catch(function(error){
            // もしsave()が失敗したらエラーをログに書き出す
            console.log(error);
        });
}
```

名前の取得を増やす

read()プログラムの変更

```
// 投稿日の日付オブジェクトを生成
var date = new Date(records[i].get("createDate"));

var time = date.getMonth() + 1 + "/"
                           + date.getDay() + " "
                           + date.getHours() + ":"
                           + date.getMinutes() + ":"
                           + date.getSeconds();
```

DBの値をJSの日付オブジェクトにする

日付オブジェクトから
表示したい文字列を作る

read()プログラムの変更

```
// 投稿内容を作成
dom.textContent = records[i].get("name")
                + ":" + records[i].get("message")
                + "(" + time + ")";
```

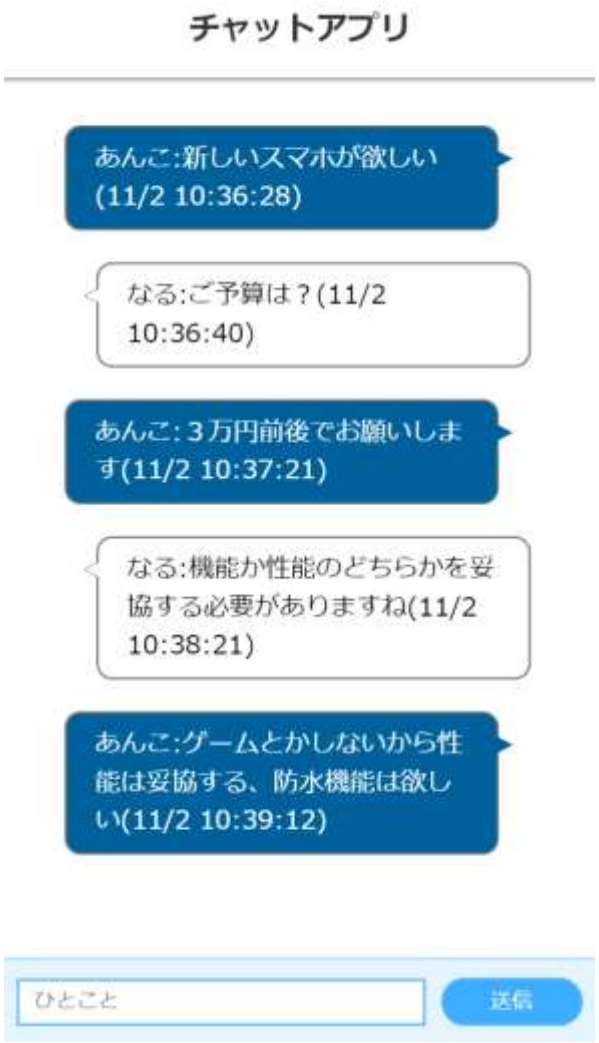
名前とメッセージと日付を
DOMに反映

チャットアプリ開発

Lv4

Lv4で実装すること

- 名前を先に設定
  - アプリ起動時に名前を聞く
- 条件分岐
  - 自分の投稿と他人の投稿でデザインを分ける



HTMLの変更

```
<form id="message_form">
    <input type="text" id="message" value="" placeholder="ひとこと">
    <input type="button" value="送信" id="sendbtn" onclick="send()">
</form>
```

名前の欄を消す

send()プログラムの変更

```
// フォームの値を取得
var message = document.getElementById("message").value;
```

名前の取得を消す

アプリの初期化処理の変更

```
// アプリの初期化処理
function init(){
    myName = prompt("なまえ");
    read();
}
```

prompt命令で名前をユーザーに入力させる

アプリの初期化処理の変更

```
// 投稿を表示するためのリストタグを生成
var dom = document.createElement("li");
// リストタグにClass属性を付与
dom.classList.add('box');
if (records[i].get("name") == myName) {
    dom.classList.add('box-right');
} else {
    dom.classList.add('box-left');
}
```

表示する投稿の名前が自分の名前なら
class「box-right」を付与

表示する投稿の名前が自分でなければ
Class「box-left」を付与

もっとカスタム版(hard)

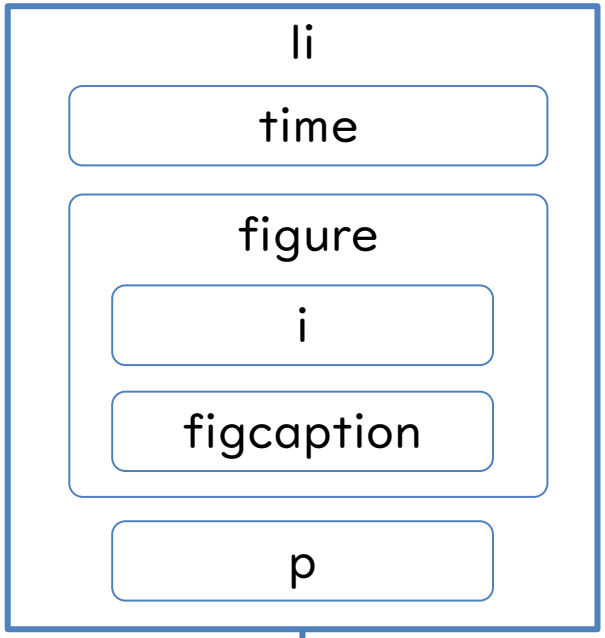
チャットアプリ開発

Lv5

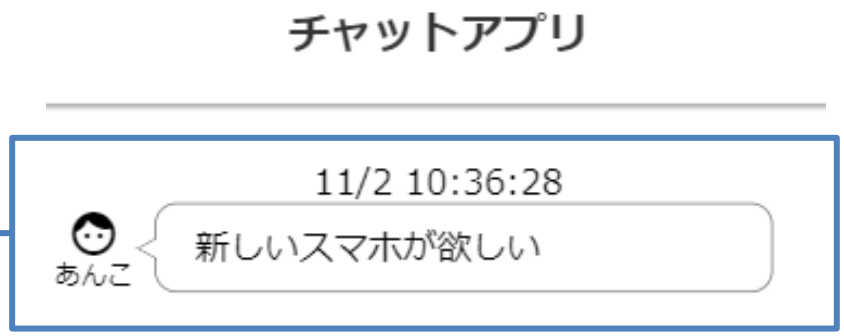
Lv5で実装すること

- 投稿内容の分割
  - 複数のタグで構成
- デザイン変更
  - アイコンが付く
  - 吹き出しになる
  - 名前はアイコンの下
  - 日時は吹き出しの上

投稿内容を複数タグで構成する

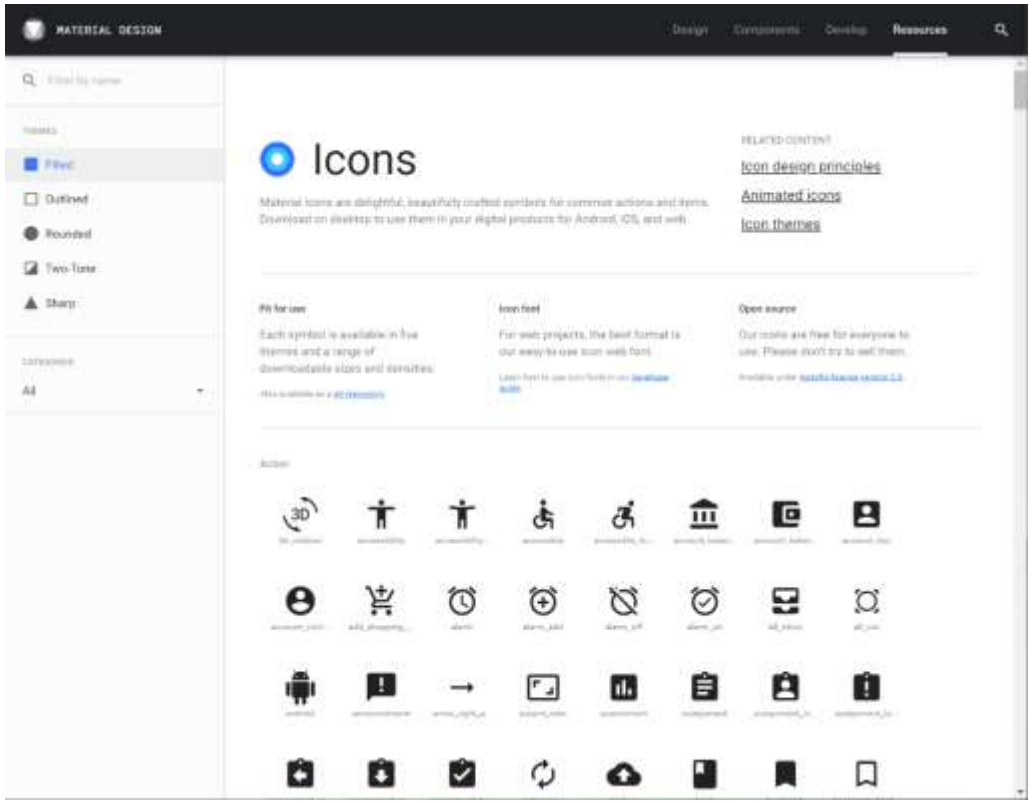



```
<li>
    <time>11/2 10:36:28</time>
    <figure>
        <i class="material-icons">face</i>
        <figcaption>あんこ</figcaption>
    </figure>
    <p class="box box-left">新しいスマホが欲しい</p>
</li>
```

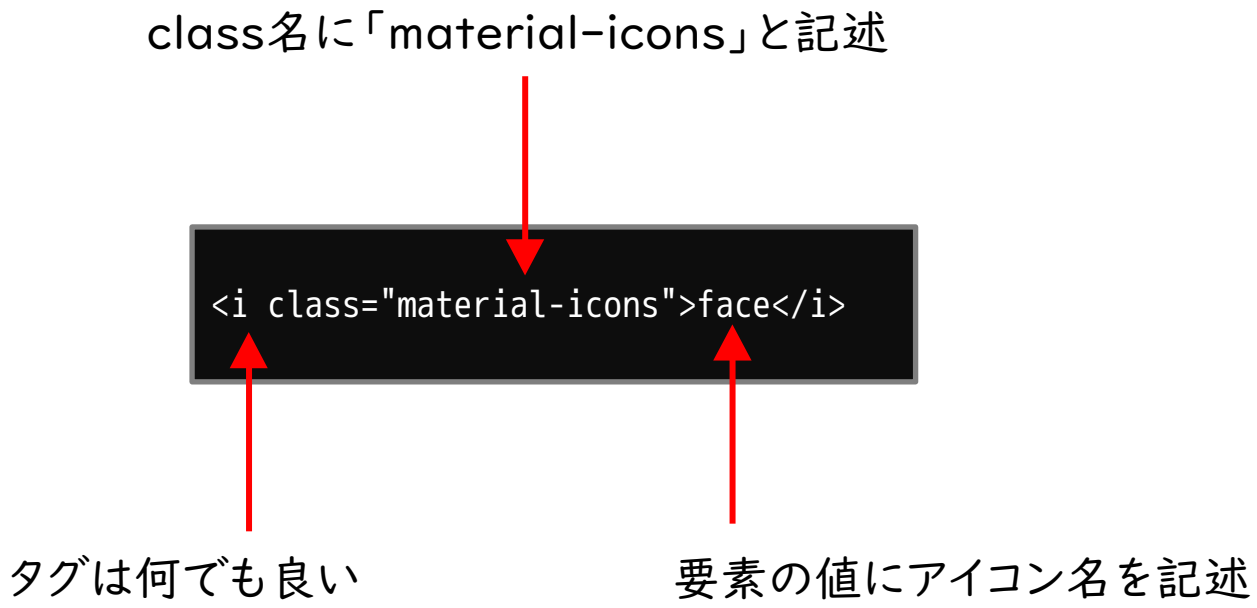


アイコンの表示について

- 「Webアイコンフォント」を利用します
    - 具体的にはGoogleの「Material Icons」



Material Iconsの使い方



Material Iconsの導入

■ HTML側のheadに一行追加すればOK

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

■ 設定イメージ

index.html × style.css

```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
       user-scalable=no">
6      <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content:
       https://ssl.gstatic.com; style-src * 'unsafe-inline'; script-src * 'unsafe-inline'
       'unsafe-eval'">
7      <script src="components/loader.js"></script>
8      <link rel="stylesheet" href="components/loader.css">
9      <link rel="stylesheet" href="css/style.css">
10     <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons" >
11     <script>
```

read()関数のfor文ループ内大幅変更（1/2）

```
for (var i = 0; i < records.length; i++) {
    // 投稿日の日付オブジェクトを生成
    var date = new Date(records[i].get("createDate"));

    var figure = document.createElement("figure");
    var icon = document.createElement("i");
    icon.textContent = "face";
    icon.classList.add('material-icons');
    var figcaption =
document.createElement("figcaption");
    figcaption.textContent = records[i].get("name");
    figure.appendChild(icon);
    figure.appendChild(figcaption);

    var time = document.createElement("time");
    time.textContent = date.getMonth() + 1 + "/"
                     + date.getDay() + " "
                     + date.getHours() + ":"
                     + date.getMinutes() + ":"
```

read()関数のfor文ループ内大幅変更 (2/2)

```
    // 投稿を表示するためのリストタグを生成
    var li = document.createElement("li");
    // 投稿内容を作成
    var p = document.createElement("p");
    p.classList.add('box');
    p.textContent = records[i].get("message");

    // 投稿内容をチャットエリアに追記
    li.appendChild(time);
    if (records[i].get("name") == myName) {
        p.classList.add('box-right');
        li.appendChild(p);
        li.appendChild(figure);
    } else {
        p.classList.add('box-left');
        li.appendChild(figure);
        li.appendChild(p);
    }
    document.getElementById("chat").appendChild(li);
}
```

チャットアプリ開発

Lv6

Lv6で実装すること

- アイコンが選べる
  - 起動時にアイコン名を聞く
- 名前とアイコンを変更できる
  - init()関数を再度呼ぶだけ
- メッセージを送らなくても
  表示を更新できる
  - read()関数を呼ぶだけ

アプリの設定の変更

```
// アプリの設定
var limit = 5;
var room = "chat";
var myName = null;
var myIcon = "face";
```

アイコン名を設定できるようにする

init()関数の変更

```
// アプリの初期化処理
function init(){
    myName = prompt("なまえ");
    myIcon = prompt("アイコン名");
    read();
}
```

アイコン名を聞く

send()プログラムの変更

```
function send() {
    // フォームの値を取得
    var message = document.getElementById("message").value;
    document.getElementById("message").value = "";
    // チャットレコードの作成と反映
    new chatTable()
        .set("message",message)
        .set("name",myName)
        .set("icon",myIcon)
        .save()
        .then(function() {
            read();
        })
        .catch(function(error){
            // もしsave()が失敗したらエラーをログに書き出す
            console.log(error);
        });
}
```

iconを保存する

read()関数のアイコン部分の小規模な変更

```
for (var i = 0; i < records.length; i++) {
    // 投稿日の日付オブジェクトを生成
    var date = new Date(records[i].get("createDate"));
    var figure = document.createElement("figure");
    var icon = document.createElement("i");

    if (records[i].get("icon")) {
        icon.textContent = records[i].get("icon");
    } else {
        icon.textContent = "face";
    }

    icon.classList.add('material-icons');
    var figcaption =
document.createElement("figcaption");
    figcaption.textContent = records[i].get("name");
    figure.appendChild(icon);
    figure.appendChild(figcaption);
```

DBにアイコンがあれば適応
なければ"face"を適応

最終カスタム版（hard）

チャットアプリ開発

Lv7

Lv7で実装すること

- 複数画面の実装
  - 今回はCSSで切り替えます
- 開始画面を付ける
  - 脱prompt()
- 複雑なフォーム対応
  - アイコンを一覧から選べる
  - チャットルームを一覧から選べる

Lv7のHTML文 1/2 ※ アイコン一覧は長いので一部省略

```
<body>
    <section id="home">
        <header>
            <h1>チャットアプリ - HOME</h1>
        </header>
        <p class="descriptive_text">チャットアプリです。マナーや法令を守って利用しましょう。</p>
        <section class="name_section">
            <input type="text" id="myName" value="" placeholder="なまえ">
        </section>
        <form id="home_form">
            <section class="icon_section">
                <ul class="icon_select">
                    <li><input name="icon" type="radio" id="icon0" value="assignment_ind"><label for="icon0"
class="material-icons">assignment_ind</label></li>
                    <li><input name="icon" type="radio" id="icon1" value="emoji_emotions"><label for="icon1"
class="material-icons">emoji_emotions</label></li>
                </ul>
             </section>
            <section class="room_section">
                <select name="room">
                    <option value="chat">チャットルーム(一般)</option>
                    <option value="chata">チャットルーム(A)</option>
                    <option value="chatb">チャットルーム(B)</option>
                    <option value="chatc">チャットルーム(C)</option>
                </select>
            </section>
            <input type="button" class="startbtn" value="開始" onclick="start()">
        </form>
    </section>
```

Lv7のHTML文 2/2

※こちらはほぼ今まで通り、mainセクション内にheaderやfooterを内包

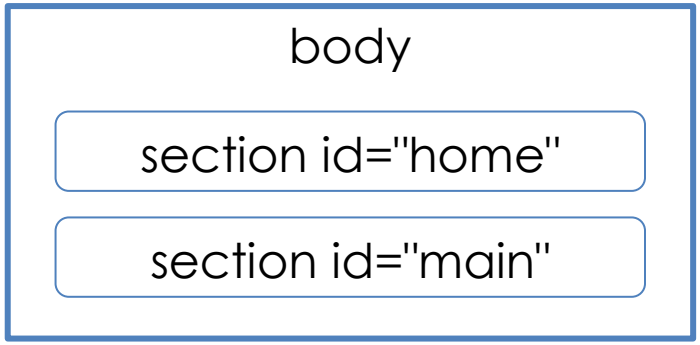
```
<body>
    <section id="main">
        <header>
            <i class="material-icons left_btn" onclick="location.reload()">home</i>
            <h1>チャットアプリ</h1>
            <i class="material-icons right_btn" onclick="read()">update</i>
        </header>
        <ul id="chat">
        </ul>
        <footer>
            <form id="message_form">
                <input type="text" id="message" value="" placeholder="ひとこと">
                <input type="button" value="送信" id="sendbtn" onclick="send()">
            </form>
        </footer>
    </section>
</body>
```

Lv7のCSS ※mainのセクションを非表示にする

```
#main {
    display: none;
}
```

body内に2つのページが存在する

- 必要に応じて表示・非表示を切り替える
  - 起動時は「home」を表示
  - 開始時は「main」を表示



新関数 start()の実装

- 役割的にはLv6までのinit()関数と似たような処理
- 主な相違点はフォームの値を変数に取り込む部分

```
// home画面のスタート処理
function start(){
    myName = document.getElementById("myName").value;
    myIcon = document.getElementById("home_form").icon.value;
    room = document.getElementById("home_form").room.value;
    chatTable = ncmb.DataStore(room);
    document.getElementById("home").style.display = "none";
    document.getElementById("main").style.display = "block";
    read();
}
```

フォームの値を取得

指定されたチャットルームで
DBテーブル選択

表示・非表示切り替え

init()関数の修正

■ 今回はやること無し、消しても良い

- 将来的に、起動画面で何か初期化処理を行いたい場合には引き続きinit()に記述する。

```
// アプリの初期化処理
function init(){
// 今回はやることなし
}
```