フロントエンド基礎 #3

Tomoyuki Takata@DeNA

勉強会の目的

開発メンバー全員がフロントエンドに対する意識を高め、
結果それがプロダクトの品質を向上させる

勉強会の流れ

1.ブラウザの仕組み、レンダリングの仕組み

2.chrome dev toolsの使い方

3.pure javascript

4.pure javascript2

5.css設計 SMACSS

6.パフォーマンスチューニング

勉強会の流れ

1.ブラウザの仕組み、レンダリングの仕組み

2.chrome dev toolsの使い方

3.pure javascript

4.pure javascript2

5.css設計 SMACSS

6.パフォーマンスチューニング

javascriptを使ってみよう！

Javascriptってどんな言語

1.**クライアントサイド/サーバーサイドどちらでも動く言語**

2.**変数に型がない**

=> 正確にはある(プリミティブ型+オブジェクト型）が他の言語ほど厳しい制約はない。

3.**同時に処理されるコードは常に1つ**

=> シングルスレッド

4.**プロトタイプ指向**

5.**関数はデータ（オブジェクト）として扱う**

Javascriptってどんな言語

backbone.js、angular.js、そして最近ではreact.jsとjavascript界隈のトレンドの移り変わりはすさまじいものがあります。これらのフレームワークは非常に便利なものですが、まずは基本の基本をしっかり押さえた上で利用しましょう！

Javascriptってどんな言語

変数とは？

変数とは、数値や文字を格納する「箱」のようなもの。
変数名とは、箱につけられた名前であり箱のなかに入ったデータを
取り出したり入れ直したりするときに利用する。

Javascriptってどんな言語

変数の宣言

var　変数名 =　値;

Javascriptってどんな言語

関数とは？

関数とは、プログラムが格納された「箱」のようなもの。
プログラムとは定められた一連の処理を実行するものであり、
プログラムが動くだけの関数もあれば、
プログラムが動いて結果を返す関数もある。

Javascriptってどんな言語

関数の定義

var　関数名 = function(){}

function 関数名(){}

Javascriptってどんな言語

関数の実行

関数名(引数);

関数名.call(scope,引数)

関数名.apply(scope,[引数])

実践1

足し算をして結果を返す関数を作ってみよう

つ　http://codepen.io/pen/

Javascriptってどんな言語

javascriptの型

- undefined
- number
- boolean
- string
- null
- object

✳number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

javascriptの型

・undefined　**未定義型**　"何のデータも定義されていない状態"を指す
・number
・boolean
・string
・null
・object

✳number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

javascriptの型

・undefined　**未定義型**　"何のデータも定義されていない状態"を指す
・number　**数値型**　正負の数値や小数も統一して扱える
・boolean
・string
・null
・object

✳number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

javascriptの型

・undefined　**未定義型**　"何のデータも定義されていない状態"を指す
・number　**数値型**　正負の数値や小数も統一して扱える
・boolean　**真偽値型**　「真（true）」と「偽（false）」の論理値
・string
・null
・object

✳number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

javascriptの型

・undefined　**未定義型**　"何のデータも定義されていない状態"を指す
・number　**数値型**　正負の数値や小数も統一して扱える
・boolean　**真偽値型**　「真（true）」と「偽（false）」の論理値
・string　**文字列型**　名前の通り文字列を扱う
・null
・object

✳number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

javascriptの型

・undefined　**未定義型**　"何のデータも定義されていない状態"を指す
・number　**数値型**　正負の数値や小数も統一して扱える
・boolean　**真偽値型**　「真（true）」と「偽（false）」の論理値
・string　**文字列型**　名前の通り文字列を扱う
・null　**Null型**　"何のデータもない状態"を指す
・object

✳number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

javascriptの型

・undefined　**未定義型**　"何のデータも定義されていない状態"を指す
・number　**数値型**　正負の数値や小数も統一して扱える
・boolean　**真偽値型**　「真（true）」と「偽（false）」の論理値
・string　**文字列型**　名前の通り文字列を扱う
・null　**Null型**　"何のデータもない状態"を指す
・object　**オブジェクト型**　JavaScript中ではすべてのオブジェクトはこの型の機能を受け継いでいる(Array,Function など)

＊number, boolean, string とかはプリミティブ値と呼ばれます
オブジェクト以外はプリミティブです。

Javascriptってどんな言語

オブジェクト型ってなに

キー(名前)と値のセットを複数持ったもの。
連想配列とかハッシュみたいなもの。

Javascriptってどんな言語

オブジェクト型ってなに

var object = {};　空のオブジェクトを定義

Javascriptってどんな言語

オブジェクト型ってなに

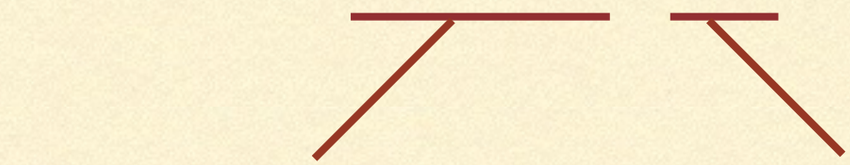
```
var object = {};
object = {test1 : 1, test2 : 2, test3 : 3};
```

Javascriptってどんな言語

オブジェクト型ってなに

```
var object = {};
```

```
object = {test1 : 1, test2 : 2, test3 : 3};
```



Javascriptってどんな言語

オブジェクト型ってなに

var object = {};

object = {test1 : 1, test2 : 2, test3 : 3 …};

キーと値のセットを複数個もつ

Javascriptってどんな言語

オブジェクト型ってなに

var object = {};

object = {test1 : “あいうえお”, test2 : 2, test3 : 3};

値は文字列だってOKだし

Javascriptってどんな言語

オブジェクト型ってなに

```
var object = {};
object = {test1  :undefined, test2 : 2, test3 : 3};
```

未定義値だってOKだし

Javascriptってどんな言語

オブジェクト型ってなに

```
var object = {};
```

```
object = {test1 :{ test4: 5, test6: 7 }, test2 : 2, test3 : 3}
```

オブジェクトだってOKだし

Javascriptってどんな言語

オブジェクト型ってなに

```
var object = {};
```

```
object = {test1 :[1,2,3,4], test2 : 2, test3 : 3};
```

配列だってOKだし

Javascriptってどんな言語

オブジェクト型ってなに

var object = {};

object = {test1 :function(){}, test2 : 2, test3 : 3};

関数だってOK

Javascriptってどんな言語

オブジェクト型から値を取り出そう

var object = {test1 :1, test2 : 2, test3 : 3};

Javascriptってどんな言語

オブジェクト型から値を取り出そう

var object = {test1 :1, test2 : 2, test3 : 3};

test1の値が欲しい場合は

```
object.test1
object[‘test1’]
```

のどちらかで取得できる

Javascriptってどんな言語

オブジェクト型に値を追加しよう

var object = {test1 :1, test2 : 2, test3 : 3};

Javascriptってどんな言語

オブジェクト型に値を追加しよう

var object = {test1 :1, test2 : 2, test3 : 3};

test4の値を追加したい場合は

```
object.test4 = 4
object[‘test4’] = 4
```

のどちらかで追加できる

実践２

以下のキーと値をもったオブジェクトを作ってみよう

キー	値
prop1	あいうえお
prop2	123
prop3	null

つ　http://codepen.io/pen/

JavascriptとDOM

javascriptを使えば、DOMをプログラムで操作することが可能。
ここではDOMエレメントの取得、追加、削除をやってみる。

JavascriptとDOM

要素を取得してみよう

あいうえお

JavascriptとDOM

要素を取得してみよう

あいうえお

jsで操作するために識別子（id）を振ります

JavascriptとDOM

要素を取得してみよう

あいうえお

getElementByIdで取得

```
var elem = document.getElementById(“test”);
```

JavascriptとDOM

要素の中身を変更してみよう

あいうえお

var elem = document.getElementById(“test”);

innerHTMLプロパティで書き換え
elem.innerHTML = “かきくけこ”;

JavascriptとDOM

要素を追加してみよう

あいうえお

JavascriptとDOM

要素を追加してみよう

あいうえお

jsで操作するために識別子（id）を振ります

JavascriptとDOM

要素を追加してみよう

あいうえお

追加するDOMを生成します

```
var childElem = document.createElement("div");
```

JavascriptとDOM

要素を追加してみよう

あいうえお

追加するDOMを生成します
```
var childElem = document.createElement("div");
```

追加するDOMの中にテキストをいれてみます
```
childElem.innerHTML = “追加するノードです”;
```

JavascriptとDOM

要素を追加してみよう

あいうえお

追加するDOMを生成します
```
var childElem = document.createElement("div");
```

追加するDOMの中にテキストをいれてみます
```
childElem.innerHTML = “追加するノードです”;
```

DOMを挿入します
```
document.getElementByld(“test”).appendChild(childElem);
```

JavascriptとDOM

要素を削除してみよう

あいうえお
    追加するノードです

DOMを削除します

document.getElementByld(“test”).removeChild(childElem);

実践3

DOMを操作してみよう

つ http://codepen.io/anon/pen/PwvpLM

Javascriptとイベント

ユーザーのアクションやページの読み込みや通信処理など,
ブラウザ上で起こるあらゆるイベントについて処理を行うのが
JavaScriptの特徴。
このようなイベントを中心としたプログラミングを
イベントドリブン（イベント駆動型）プログラミングと呼ぶ。

Javascriptとイベント

addEventListenerを理解しよう

Javascriptとイベント

addEventListenerを理解しよう

window.addEventListener(‘load’,関数,false);

Javascriptとイベント

addEventListenerを理解しよう

window.addEventListener(‘load’,関数,false);

監視しているオブジェクト＝イベントターゲット

Javascriptとイベント

addEventListenerを理解しよう

window.addEventListener(‘load’,関数,false);

監視しているイベント

Javascriptとイベント

addEventListenerを理解しよう

window.addEventListener(‘load’,関数,false);

監視しているイベントが起こったときに
呼ばれる関数＝イベントリスナー

Javascriptとイベント

addEventListenerを理解しよう

window.addEventListener(‘load’,関数,false);

ページが完全に読み込まれた時に関数を実行しなさい

＊IE9より前の Internet Explorerでは標準の addEventListener ではなくattachEvent を使わなければならない

Javascriptとイベント

様々なイベント

element.addEventListener(“mouseover",関数,false);

element.addEventListener("click", 関数,false);

element.addEventListener("touchstart", 関数,false);

Javascriptとイベント

様々なイベント

element.addEventListener(“mouseover",関数,false);
=> 要素の上にマウスが乗った際に呼ばれる
element.addEventListener("click", 関数,false);

element.addEventListener("touchstart", 関数,false);

Javascriptとイベント

様々なイベント

element.addEventListener(“mouseover",関数,false);
=> 要素の上にマウスが乗った際に呼ばれる
element.addEventListener("click", 関数,false);
=> 要素がクリックされた際に呼ばれる
element.addEventListener("touchstart", 関数,false);

Javascriptとイベント

様々なイベント

element.addEventListener(“mouseover",関数,false);
=> 要素の上にマウスが乗った際に呼ばれる
element.addEventListener("click", 関数,false);
=> 要素がクリックされた際に呼ばれる
element.addEventListener("touchstart", 関数,false);
=> 要素に指が乗った際に呼ばれる（モバイル）

実践４

要素をクリックしてアラートを出してみよう

つ http://codepen.io/anon/pen/dPEvze

Javascriptとイベント

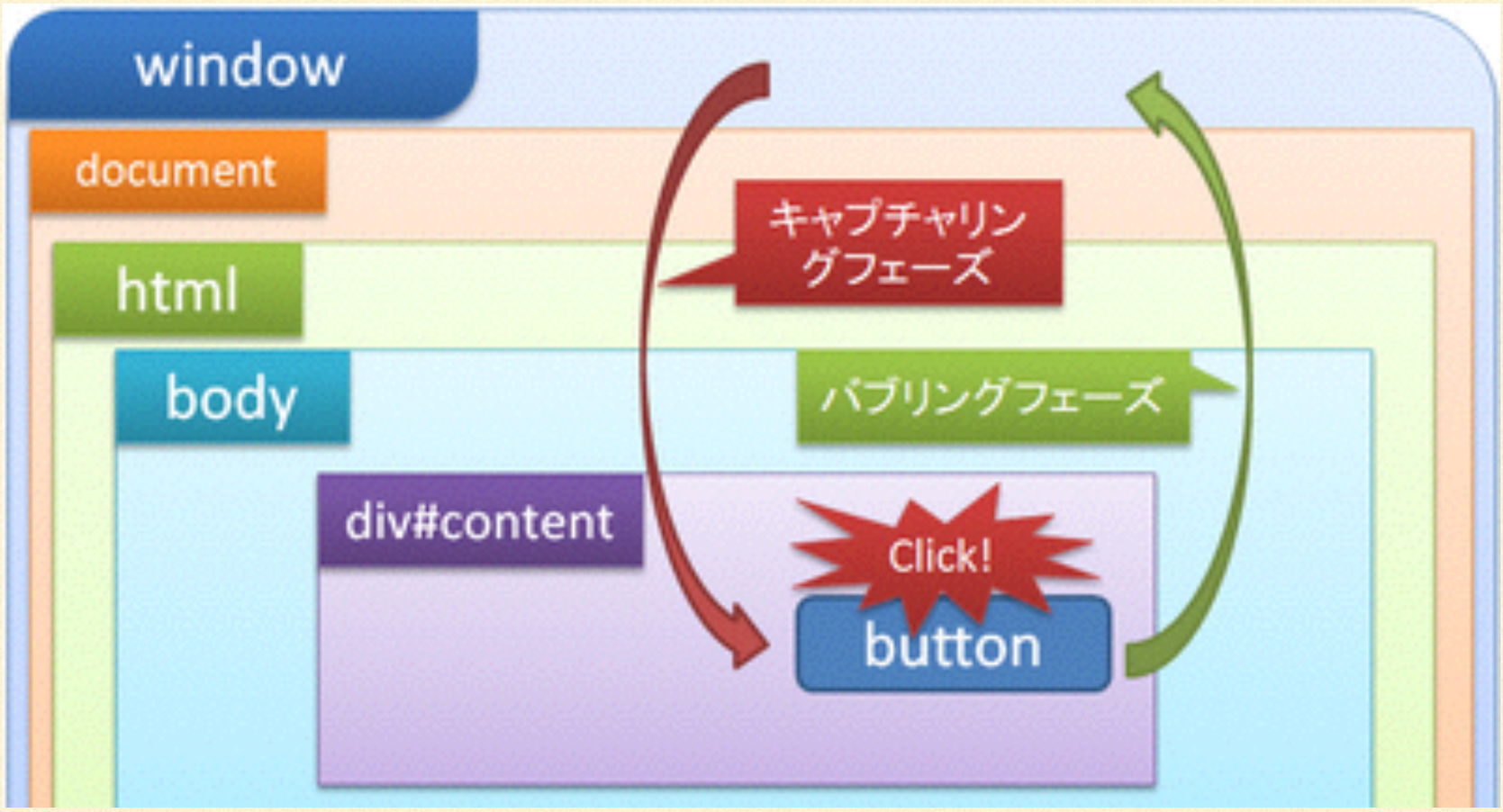
addEventListenerを理解しよう

window.addEventListener(‘load’,関数,false);

そういえばこの引数はなに？

Javascriptとイベント

addEventListenerを理解しよう



useCapture
をfalse

次回へ続く