

Presentation
for

JS選手権

Aira Sakajiri

2018.11.15

なぜプログラミングを
難しいと感じる？



プログラミング言語



I Have a Dream

私には夢がある





!

■

JSの翻訳機を作ろう



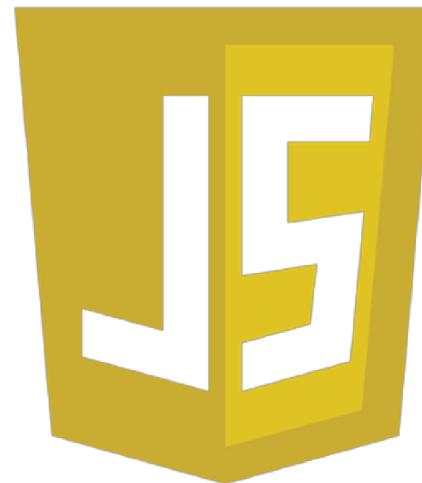
(プログラム)
Grammar & Graphic

HTML



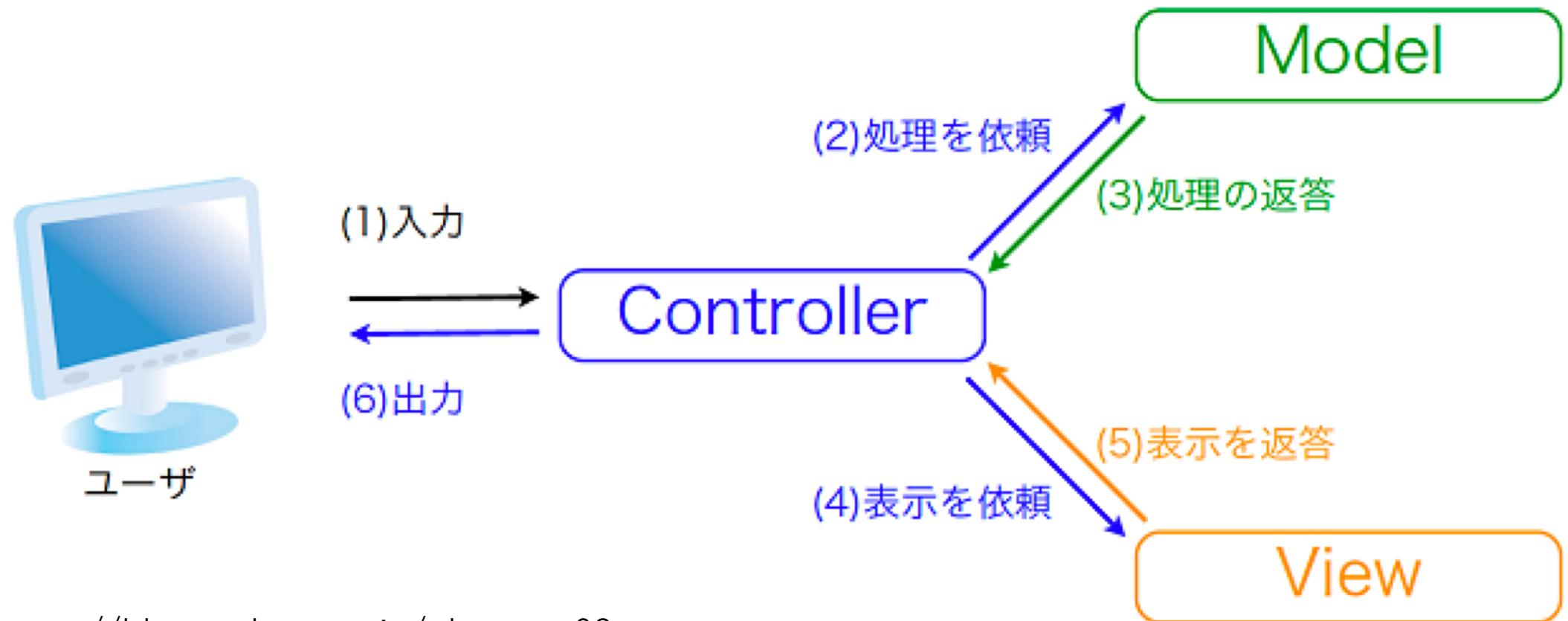
Sass

JS



Cloud
Firestore

MVCモデルに基づく サイト構築を実現(w/o フレームワーク)



```
GrammerController.js x JS GrammerModel.js JS Gramme 🔍 ⚡ ...  
ackathon › js › modules › controller › JS GrammerController.js › ...  
controlStatements Aa Ab! * 結果なし ← → ⌂ x  
import GrammerModel from "../model/GrammerModel.js";  
import GrammerView from "../view/GrammerView.js"  
import GraphicView from "../view/GraphicView.js"  
  
export class TranslateController{  
  constructor(props){  
    console.log('props : ' + props.for);  
    // this.codes = props;  
    this.bindMethods();  
    this.init();  
  
    // const result = this.gm.translateToLanguage(this.codes);  
    // this.gv.appendText(result);  
  
    this.translateEachCords();  
    this.gpv.appendVariable(this.gmm.variableObj);  
  }  
  
  init(){  
    this.gmv = new GrammerView({  
      onSendCallBack: this.translateEachCords  
    });  
    this.gmm = new GrammerModel();  
    this.gpv = new GraphicView({  
      onSendCallback: this.drawCords  
    });  
  }  
  
  bindMethods(){  
    this.translateEachCords = this.translateEachCords  
    this.drawCords = this.drawCords.bind(this);  
  }  
  
  /*  
   * Grammer Controller  
   */  
}  
  
JS GrammerModel.js x ...  
js-hackathon › js › modules › model › JS GrammerModel.js › GrammerModel › ...  
1 import VariableModel from "./VariableModel.js";  
2  
3 export class GrammerModel{  
  constructor(props){  
    // this.codes = props.codes;  
    // this.code = this.codes; //複数行にするまでの繋ぎ  
    this.init();  
  }  
  init(){  
    this.mainLineNum = [];  
    this.variableObj = {};  
    this.functionObj = {};  
    this.controlStatements = [];  
    this.layerNumber;  
    this.isCommentFlag = false;  
    this.translatorUrl = 'https://script.google.com/m';  
  }  
  
  /*-----  
   * Translator  
   -----*/  
  adjustFuncNameToText(funcName){  
    funcName = funcName.split(/([A-Z][a-z]*)/).filter(  
      el => el == '' || el == undefined) return false;  
    return true;  
  };  
  funcName.forEach((el,i)=>{  
    //Topは大文字、それ以外は小文字に  
    if(i==0){  
      funcName[i] = el.charAt(0).toUpperCase()  
    }else{  
      funcName[i] = el.charAt(0).toLowerCase()  
    }  
  });  
  funcName = funcName.join(' ')  
  console.log(`関数 ${funcName} を翻訳します...`);  
  return funcName;  
}  
translateText(thisText,targetLang){  
  /*  
   * Grammer Model  
   */  
}
```



```
JS GrammerView.js x ...  
js-hackathon › js › modules › view › JS GrammerView.js › GrammerView › ...  
1 export class GrammerView{  
  constructor(props){  
    // this.code = props.code;  
    this.onSendCallBack = props.onSendCallBack;  
    //console.log(this.onSendCallBack);  
    this.init();  
  }  
  
  init(){  
    this.loadAceSettings();  
    this.bindDomElement();  
    this.handleEvents();  
  }  
  
  getEditorString(){  
  }  
  
  bindDomElement(){  
    this.$grammer = $('#grammer');  
  }  
  
  handleEvents(){  
    this.editor.getSession().on('change', (e)=>{  
      if(e.lines[0]==';'){  
        console.log('[Comment];の入力を検知=====');  
        this.onSendCallBack();  
        return false;  
      }  
    });  
  
    $(window).keydown((e)=>{  
      if(event.shiftKey){  
        if(e.keyCode == 13){  
          console.log('[Comment]Shift + Enterの入力を検知=====');  
          this.onSendCallBack();  
          return false;  
        }  
      }  
    })  
  }  
}
```

翻訳ロジックの作成

let or const?

i = i + 1 の計算結果は？

i += 1 の算出方法は？

どこまでが function…？

コンソールログによる ログ情報の可視化



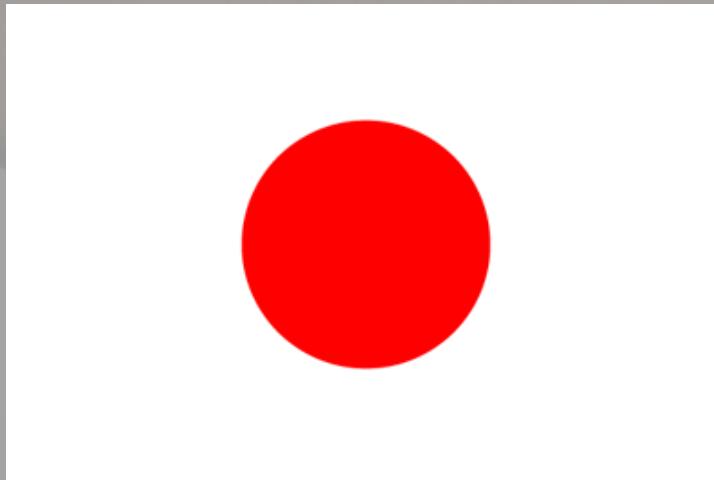
The screenshot shows the browser's developer tools with the 'Console' tab selected. The output area displays a series of log messages from the 'GrammerController.js' file, illustrating how log statements are printed to the console.

```
props : ctrlsta GrammerController.js:7
[Message] メッセージの内容を削除します===== GrammerView.js:45
[Message] 関数情報を取得します。===== GrammerController.js:46
関数 printOnConsoleByThisFunction を発見しま GrammerModel.js:79
した。
printOnConsoleByThisFunction の情報取得が GrammerModel.js:116
完了しました。
▶ Object GrammerModel.js:117
END GrammerController.js:51

[Message] 制御構文の有無を確認します===== GrammerController.js:53
関数 printOnConsoleByThisFunction を発見しま GrammerModel.js:79
した。
printOnConsoleByThisFunction の情報取得が GrammerModel.js:116
完了しました。
▶ Object GrammerModel.js:117
END GrammerController.js:59

[Message] 実際の計算を開始します。===== GrammerController.js:61
- [開始]1行目の処理 GrammerController.js:64
処理対象コード: GrammerModel.js:182
- [終了]1行目の処理 GrammerController.js:91
- [開始]2行目の処理 GrammerController.js:64
処理対象コード: GrammerModel.js:182
- [終了]2行目の処理 GrammerController.js:91
- [開始]3行目の処理 GrammerController.js:64
処理対象コード: GrammerModel.js:182
/*
これはコメント文です。 GrammerModel.js:184
- [終了]3行目の処理 GrammerController.js:91
- [開始]4行目の処理 GrammerController.js:64
処理対象コード: GrammerModel.js:182
';'を入力 or Shift+Enter キーで
これはコメント文です。 GrammerModel.js:184
- [終了]4行目の処理 GrammerController.js:91
- [開始]5行目の処理 GrammerController.js:64
```

Localization



プログラミングをもっと身近に。

(プログラム)
Grammar & Graphic