



НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Въведение в Bootstrap

- готова обвивка (framework) за фронт-енд решения
- с него получаваме наготово:
 - css reset
 - css база за изграждане на responsive design (grid система, цветове, фонтове и т.н.)
 - предварително изготвени компоненти (като формуляри, диалогови прозорци, навигационни панели и т.н.)
 - javascript функционалност

Полезни връзки

- Официална страница
 - <http://getbootstrap.com/>
- Демонстрация на Bootstrap компонентите:
 - <https://blackrockdigital.github.io/startbootstrap-sb-admin-2/pages/index.html>
- Bootstrap cheatsheet:
 - <http://hackerthemes.com/bootstrap-cheatsheet>

Grid система, основни класове

- Грид системата използва 12 деления на екрана
- Основни класове за layout:
 - **container**: използва се за основното съдържание на страницата, като го оразмерява и центрира (responsive)
 - **row**: групира колоните от грид системата (различните col-xx-x)
 - **col-SIZE-WIDTH**, където
 - SIZE е едно от: xs, sm, md, lg
 - WIDTH е число от 1 до 12

Елементи и цветове, ОСНОВНИ КЛАСОВЕ

- Bootstrap ни предоставя наготово класове, които задават определени стилове на нашите елементи, оформят ги и им задават цветова база
- jumbotron, page-header
- btn, btn-lg, btn-sm
- panel, panel-heading, panel-body
- *-default, *-info, *-success, *-warning, *-danger

Стилизиране на формуляри с Bootstrap

- form-horizontal
- form-inline
- form-group
- input-group
- form-control

Примери

- <http://getbootstrap.com/css/#forms>
- http://www.w3schools.com/bootstrap/bootstrap_forms_inputs.asp
- http://www.w3schools.com/bootstrap/bootstrap_forms_inputs2.asp
- <http://getbootstrap.com/components/#input-groups>
- <https://v4-alpha.getbootstrap.com/components/forms/>

Да се запознаем с:



Java Script

Събития и обработка

HTML атрибути за събития

- `onclick=""`
- `onmouseover=""`
- ...

JavaScript функции за обработка

- `alert()`
- `confirm()`
- ...

JS Plugins & Bootstrap.js

JavaScript Plugins Repository

<https://plainjs.com/javascript/plugins/>

Bootstrap.js

- modals (dialogs) | <http://getbootstrap.com/javascript/#modals>
- tooltips | <http://getbootstrap.com/javascript/#tooltips>
- accordion | <http://getbootstrap.com/javascript/#collapse-example-accordion>
- affix | http://www.w3schools.com/bootstrap/bootstrap_affix.asp
- carousel | <http://getbootstrap.com/javascript/#carousel>
- tabs | <http://getbootstrap.com/javascript/#tabs>

JQuery & JQuery UI

- JavaScript е език за програмиране
- JQuery е надграждане над JavaScript, написано на JavaScript
- JQuery е JavaScript framework
- JQuery ни дава по-богата семантика спрямо native JS
- JQuery е много по-удобен за ползване, но истинските FE ninjas могат да напишат всичко и на native JS
- JQuery UI е колекция от JS плъгини, написани на JQuery

Фреймуърк?

- Bootstrap е CSS/JS фреймуърк (framework)
- JQuery е JavaScript библиотека и framework
- И двата framework-а целят да ни дадат готови инструменти, с които да можем да работим бързо, лесно и безпроблемно
- Има много JS плъгини, базирани на JQuery, като повечето от тях, могат да бъдат открити тук:
 - <https://plugins.jquery.com/>

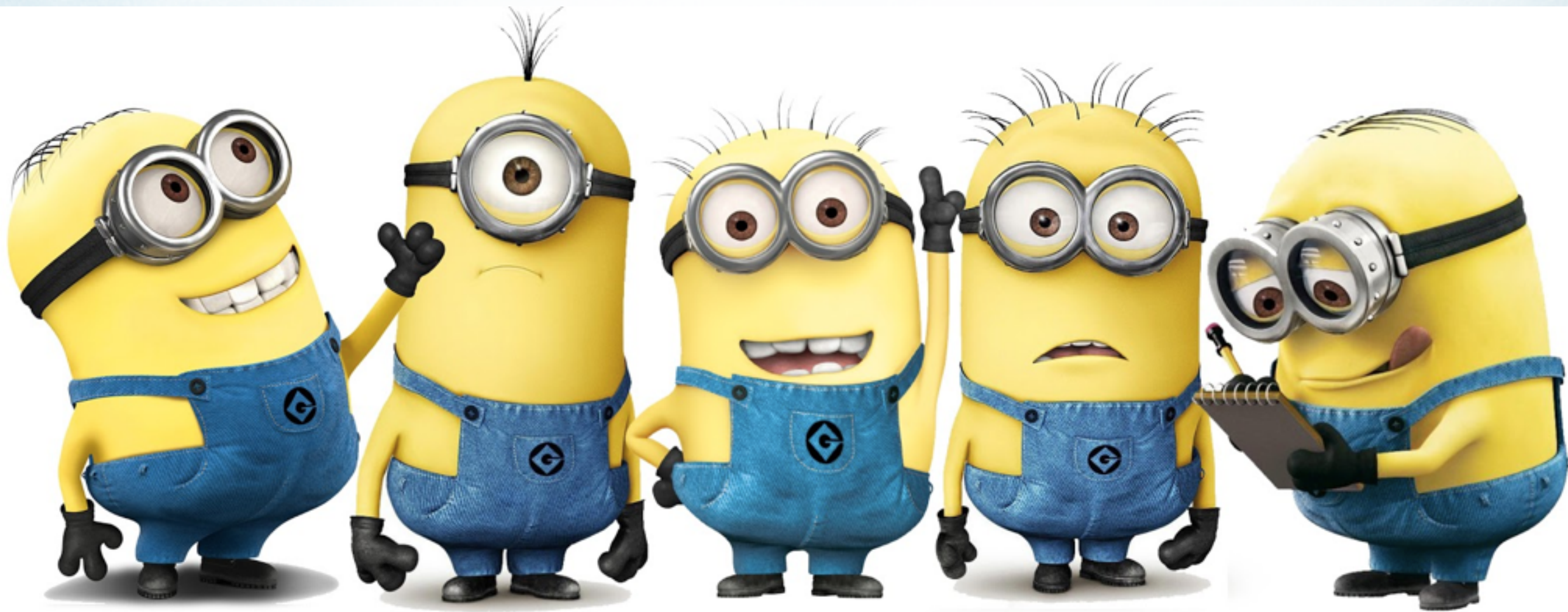
With framework:



Without framework:



Програми и езици за програмиране



Програмиране

- Програмирането е процес на съставянето на програма или *приложение*, посредством език за програмиране
- Езиците за програмиране са изразно средство, близко до човешкия език, чрез което можем да изградим логиката и поведението на нашата програма
- Езика за програмиране се състои от:
 - набор от ключови думи (за различните команди и операции)
 - семантика (какво представляват командите и как се ползват)
 - граматика и синтаксис (как се изписват ключовите думи и как се организират тези ключови думи в инструкции)

Програма

- Най-просто казано, програмата е набор от инструкции, които се изпълняват в определена последователност и работят с определени данни (стойности)
- В общия случай програмата има за цел да даде някакъв резултат на базата на входните данни и направените върху тях изчисления



Пример:

Проверка дали потребителско име и парола са коректни

- **ВХОДНИ ДАННИ:**

стойността на потребителското име, стойността на паролата

- **ИЗЧИСЛЕНИЯ:**

проверка дали в базата данни са записани същите стойности като подадените

- **РЕЗУЛТАТ:**

valid / invalid

Още примери:

- Програма на пералната машина:
 - това са различните комбинации от функциите на машината: *пране, предпране, изплакване, градуси* и т.н.
- Програма на кафе-машината:
 - входните данни са *стотинките, които пускаме и бутона, който натискаме*
 - резултата може да бъде: *очакваното кафе* или *грешка* (не сме въвели необходимата сума, свършило е млякото, няма ресто и т.н.)
- И в двата случая се използват програми, които да изпълняват последователно поредица от инструкции, при това - по много пъти

Приложение

- Приложение (*application*) наричаме крайният (*завършен*) продукт, който сме изградили чрез софтуерните програми и езиците за програмиране
- Това е софтуерен продукт, насочен към някакъв краен потребител (*клиент*)
- Приложенията могат да правят много неща и да работят с много данни наведнъж. Те са съвкупност от по-малки програмни "единици" като например: *логин на потребител*
- Уебсайтовете са *web приложения*

Back-end & Front-end

- Най-често един уебсайт се състои от 2 основни части:
 - *back-end* приложение (API)
 - *front-end* приложение (client)
- *Back-end* приложението е необходимо тогава, когато трябва да се ползват съхранени данни (като например потребителски данни, данни за продукти и т.н.)
- Всяко приложение си има клиент
 - клиента на *back-end* приложението е *front-end* приложението
 - а клиента на *front-end*-а е .. браузърът, разбира се :)

Поведение на приложението

- В зависимост от получените входни данни и от настъпилите събития (изменения) по време на изпълнение на програмата, тя може да завърши по различен начин и да даде различен резултат
- Логиката, за това как действа програмата в различните случаи, се нарича поведение или *behaviour*
- Тази логика се определя от т.нар. product owner, т.е. собственика на бизнеса, който съответното приложение обслужва
- Пример - бизнесмен и идва при вас със следната поръчка:
Искам сайт за новини, който да има по 5 новини в делничните дни и по 2 през уикенда

Заключение

- Програмата работи с данни, върху които прави изчисления и логически операции, с цел получаване на някакъв резултат или ефект
- За да съставим програма, ни е необходимо да знаем предварително само следните неща:
 - какво искаме да прави (какъв е желаният резултат)
 - какви са входните данни
 - какви инструкции и в каква последователност трябва да използваме, за да получим желанния резултат
- Програмата ще свърши нещо вместо нас и ще ни спести работа
- Програмата ще може да се използва многократно



And now you know what programming is

Въпроси?

Примери

<http://swift-academy.zenlabs.pro/lessons/lesson11/examples/download.zip>

Домашно

<http://swift-academy.zenlabs.pro/lessons/lesson11/homework>