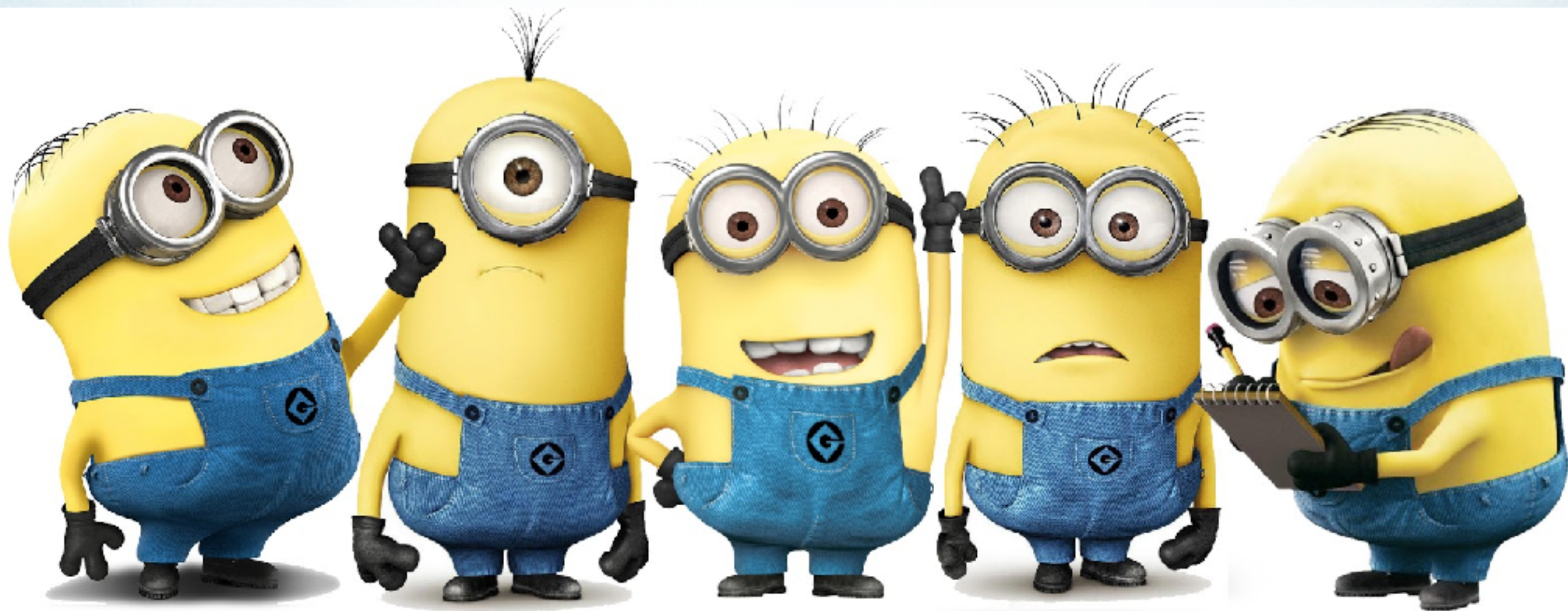




НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

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 желания 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

Да се запознаем с:



Java Script

Събития и обработка

HTML атрибути за събития	JavaScript функции за обработка
<ul style="list-style-type: none">• <code>onclick=""</code>• <code>onmouseover=""</code>• ...	<ul style="list-style-type: none">• <code>alert()</code>• <code>confirm()</code>• ...

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>

Въпроси?

Примери

<http://swift-academy.zenlabs.pro/lessons/lesson12/examples/download.zip>

Домашно

<http://swift-academy.zenlabs.pro/lessons/lesson12/homework>