



НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

# Cookies





*Sometime Me Think, 'What is Friend?' and then me say, 'Friend is someone to share the last cookie with.'*

*– Cookie Monster*

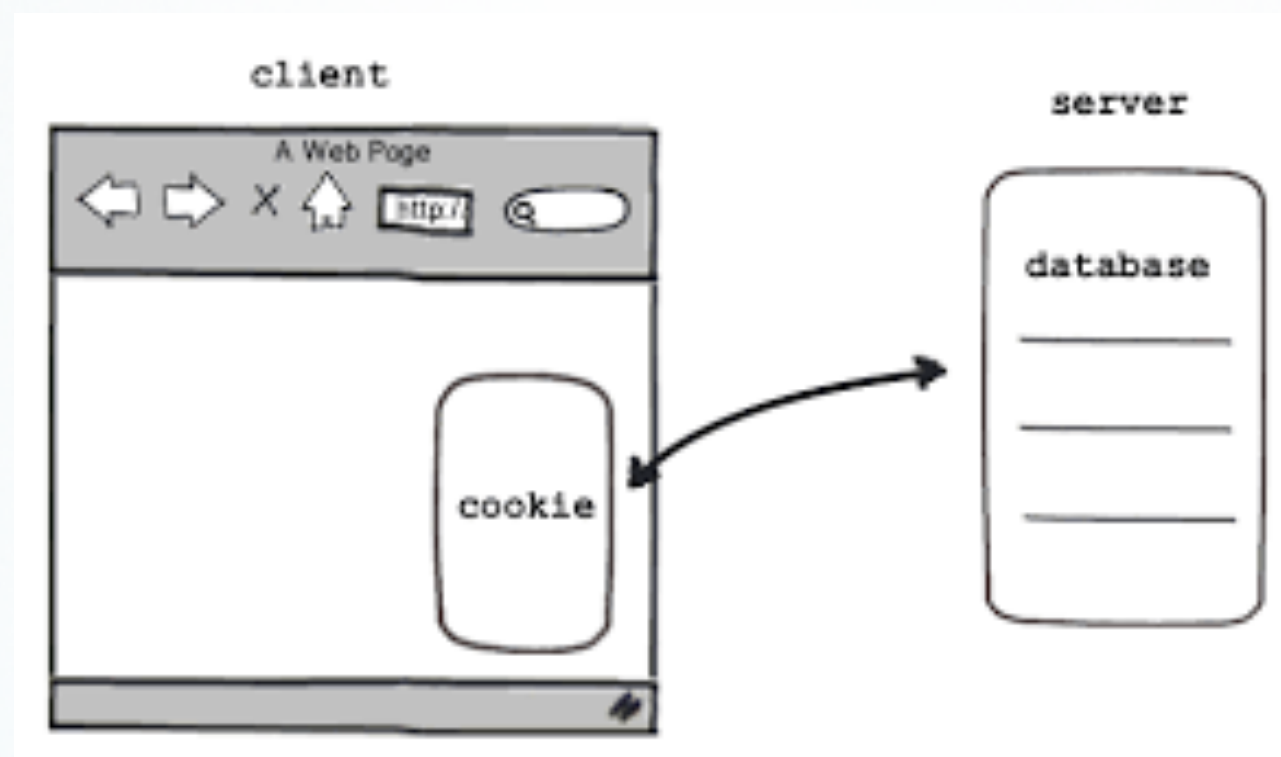
# Какво са и защо ги има?

- Бисквитките са малки текстови файлове на компютъра на клиента, които използваме за да си пазим в тях текущото състояние на страницата ни
- Тъй като един потребител може да отваря страницата от най-различни компютри и устройства, всяко едно от тези устройства помни дали потребителя се е логнал, дали е избрал език и т.н.
- Ако тази информация се пазеше на сървъра, това би означавало че ако аз се логна в дадена страница, след това, всеки който я отваря да е логнат в моя профил
- <http://www.webopedia.com/TERM/C/cookie.html>
- <http://www.whatarecookies.com/>



# Защо са толкова важни

- Както казахме, в бисквитката се пази сесия на потребителя - тя помага да се запомнят нещата, които той е направил по страницата
- Т.е. в бисквитката се съхранява само това, което е важно и ще се ползва единствено от текущия потребител, а в базата на сървъра - всичко останало



# Как се прави бисквитка

- Бисквитката може да се създаде и във front-end-a и в back-end-a, но по-често това става в back-end-a
- Класическият случай е когато имаме логин на потребители:
  - При получаване на username и парола и при успешен логин, сървърът прави бисквитка, в която слага ID на сесията на потребителя
  - След това я връща на клиента и той я запазва на компютъра на потребителя.
  - От тук-нататък всеки път, когато клиента прави заявка, трябва да изпраща и бисквитката (иначе сървърът може да му откаже достъп до някои данни, например лични данни като кредитни карти, сметки и т.н.)
- Предаването на бисквитката от сървъра към клиента и от клиента към сървъра, става като бисквитката се включва в HTTP хедърите на съобщението (т.е. на HTTP request-a/response-a)

HTTP  
Client

HTTP  
Server

Login  
POST  
username=david  
password=davidh

Login successful?

1. create session id
2. return session id in cookie
3. store session id in database

**SESSION ID**  
Sessionid  
Username  
createDate  
expireDate  
lastAccessDate

Set-Cookie: SESSIONID=66C530ACAF44D1605588619ECB0C737C

HTTP is Stateless

Cookie: SESSIONID=66C530ACAF44D1605588619ECB0C737C

Lookup Session ID

1. session match a username?
2. session still valid?

Content for 'david'

Database





# Как изглеждат бисквитките?





# document.cookie

```
document.cookie = "name=Jeni; expires=Thu, 18 Mar 2017 12:00:00 UTC; path=/";
```

↓  
**КЛЮЧ И СТОЙНОСТ**

↓  
**ДО КОГА Е ВАЛИДНА**

↓  
**КЪДЕ ЩЕ СЕ  
СЪХРАНЯВА ФАЙЛА**

Ако създадем бисквитката без `expires` или `path` (т.е. `document.cookie = "name=Jeni";`) то тя ще се създаде със срок и път по подразбиране, но ние няма как да знаем какви са техните стойности

# Expiration time

- Бисквитките имат срок на годност
- Ако браузърът види че една бисквитка е "изтекла", я изхвърля (изтрива)
- Срока на бисквитката се задава при създаването ѝ:

```
document.cookie = "name=Jeni; expires=Thu, 18 Mar 2017 12:00:00 UTC; path=/";
```



expiration time



# Промяна и изтриване

- След като сме създали една бисквитка, можем да променим стойността ѝ, като просто отново я зададем на `document.cookie`:

```
document.cookie = "name=Pepi; path=/";
```

- Тъй като вече има създадена бисквитка с ключ `name`, (и път '/') това просто ще ъпдейтне стойността ѝ
- За да изтрием бисквитката, просто трябва да я зададем отново на `document.cookie`, но този път със стойност на `expire` - в миналото:

```
document.cookie = "name=; expires=Thu, 01 Jan 1970 00:00:00 UTC";
```

или:

```
document.cookie = "name=; expires=" + (new Date(0)).toString();
```

# Въпроси?



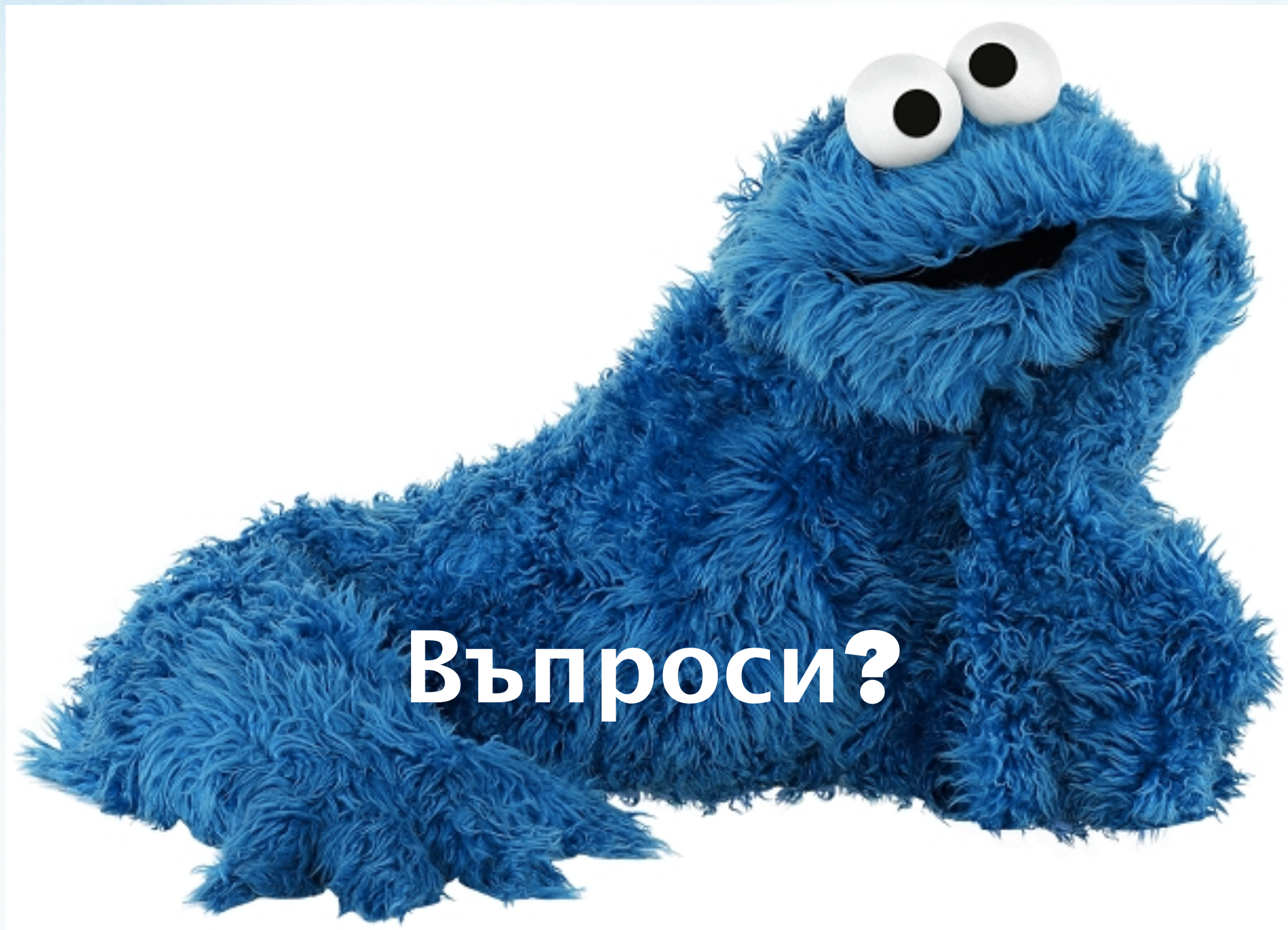
# Примери



# *Защо някои уебсайтове питат дали може да ползват бисквитки?*

- Защото бисквитката е съдържание, което идва от интернет и се съхранява на моя компютър
- Може да реша че не искам да разрешавам подобно нещо
- При това положение обаче няма да мога да ползвам и страницата
- Подобно е на забраняването на JavaScript
- И все пак - прието е да се пита първо :)





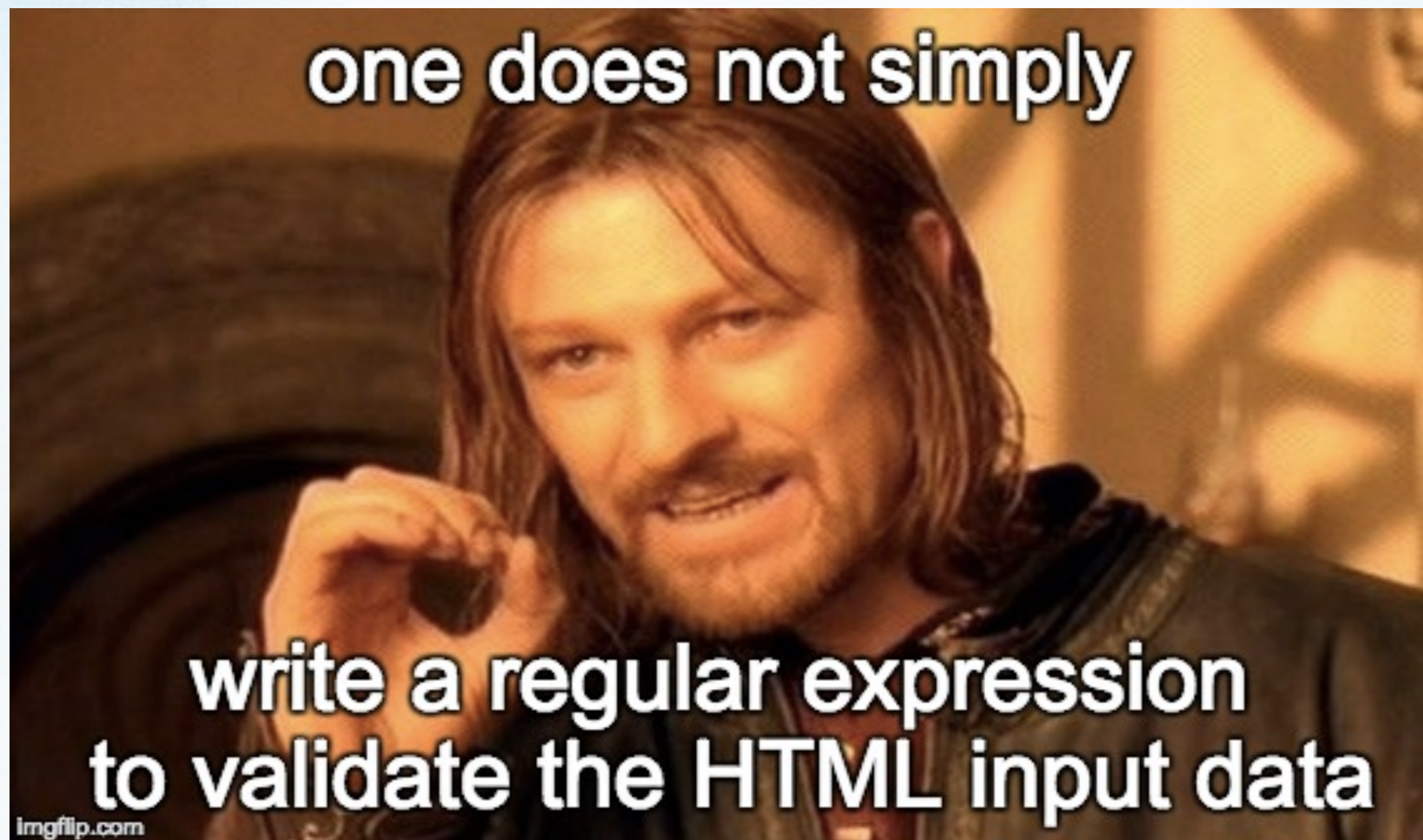


# Полезни връзки (Cookies)

- <https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie>
- <https://github.com/carhartl/jquery-cookie>
- [http://www.w3schools.com/js/js\\_cookies.asp](http://www.w3schools.com/js/js_cookies.asp)



# RegEx





# Какво представляват регулярните изрази

- Това са шаблони за текст
- В тях използваме неща като "\*" или други специални знаци, на местата където искаме да ни съвпадат повече от една букви от текста

- Например:

шаблона

`/how are you.*`

ще съвпадне с:

"how are you" и с "how are you?", и с "how are you my friend?"

# И още примери

- шаблона: `/I love .*/`
- ще съвпадне с:  
"I love pizza!" и с "I love ponies!"
- както и със:  
"I love pizza and ice cream, and football, and beer!"
- но няма да съвпадне със:  
"I don't love baseball!", "I like blues!" и "I LOVE you!"

*Т.е.: буква съвпада с буква (малка - с малка, голяма - с голяма), интервал съвпада с интервал, а `.` съвпада със всичко*

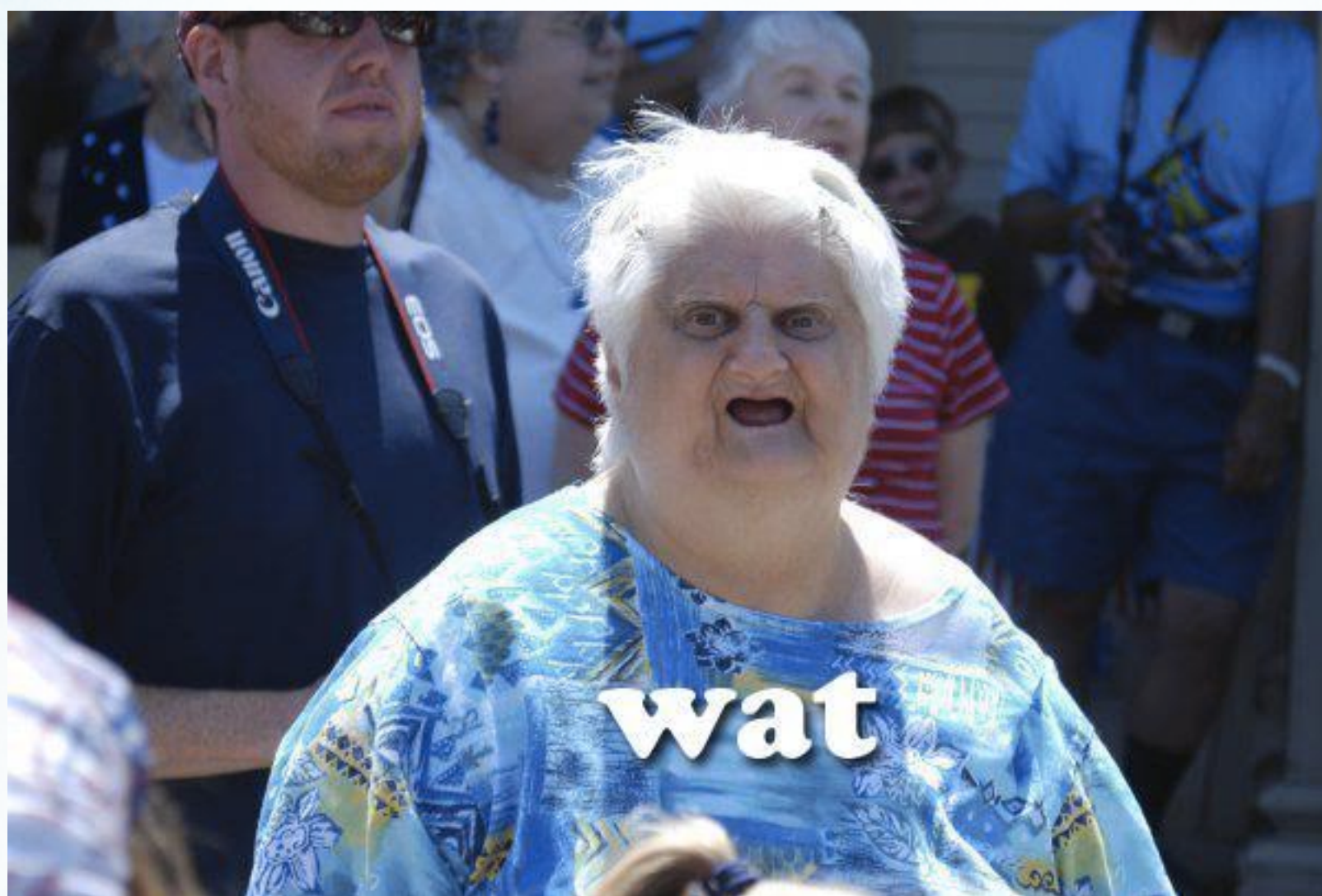


# Приложение на RegEx

- Регулярните изрази ни помагат да проверим дали даден текст "пасва" на даден шаблон (pattern matching)
- шаблона може да проверява за най-различни неща - от това дали стринга се състои само от букви или само от цифри, до това колко да е дълъг, какви символи да има в него и на какви позиции
- Съответно, за да проверим дали един email е валиден, трябва да си направим шаблон от типа на:

<текст>@<текст>.<текст>

- нека пробваме ето така: `email.match(/w+@w+\.w+/)`





# Как да разгадаем тези кодове и знаци?

- Основното, което трябва да знаем е:
  - регулярният израз се изписва между две наклонени черти:  
**`/the-regex-comes-here/`**
  - Всеки интервал или препинателен знак има значение!!  
**`/match/ !== / match /`**
  - Символите: `!, ?, ., -, [, ], (, ), {, }, |, *, +, ^, $, \` и `/` са специални и се използват за маркиране на различни видове текст в стринг-а
  - Специалният символ `"\"` се използва за различни видове селектори (на букви, числа и т.н.) като `\w`, `\W`, `\d`, `\D` и т.н., както и за escape-ване на всеки един от специалните символи.  
Например, за да match-нем `"/"`, трябва да напишем: `/\/`

# Селектори

- За да получим съвпадение с текст, трябва да укажем в регулярния израз какви видове символи искаме да match-нем
- За целта можем да ползваме range селектора: []  
Например: `/[1-9]/` ще match-не текст, който съдържа цифра от 1 до 9
- Също така имаме на разположение и няколко специални селектора:
  - `\d` - мачва точно една цифра от 0 до 9 (`/\d/` === `/[0-9]/`)
  - `\w` - съвпада с точно една буква, цифра или `_` (`/\w/` === `[a-zA-Z0-9_]`)
  - `\s` - съвпада с точно един **white space** символ: space ( ), нов ред (`\n`) или tab (`\t`)



# Quantifiers

- Можем да кажем колко пъти искаме даден selector да се повтаря в текста, като използваме следните знаци: ?, +, \* и {}. Например:
  - `\d?` - ще match-не точно нула или една цифра
  - `\d+` - ще match-не точно една или повече цифри
  - `\d*` - ще match-не точно нула или повече цифри
  - `\d{2}` - ще match-не точно 2 цифри
  - `\d{3,5}` - ще match-не точно от 3 до 5 цифри

# Примери

```
mobilePhone.match (/^ (\+|00) 359 ?\d{3} ?\d{3} ?\d{3}$/)
```

```
birthDate.match (/^\d\d\d\d-\d\d-\d\d$/)
```

```
birthDate.match (/^(19\d\d|20(0\d|1[0-5]))-[01]?\d-[0123]?\d$/)
```

```
cardNumber.match (/^\d{13}$/)
```

```
name.match (/^[A-Z][a-z]+ [A-Z][a-z]+$/)
```

```
url.match (/^https?:\/\/\w[\w+\.\w/]+$/)
```

```
email.match (/^\w[\w.-]+@(\w+\.)+\w+$/)
```



# Домашно (RegEx)

Да се направи регистрационна форма, която да има следните полета:

name, email, password, password confirmation, birthdate, picture url

Направете Javascript валидация за всички полета, като използвате регулярни изрази.

# Полезни връзки (RegExp)

- [https://developer.mozilla.org/en/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en/docs/Web/JavaScript/Guide/Regular_Expressions)
- <https://www.debuggex.com/cheatsheet/regex/javascript>
- <https://regex101.com/>



# Примери

(RegEx & Cookies)

<http://zenlabs.pro/courses/lessons/lesson19/examples.zip>

<http://codepen.io/jenie/pen/qNExYg?editors=1010>

<http://codepen.io/jenie/pen/MeYGjW?editors=1011>

# Домашно

<https://github.com/zzeni/swift-academy-homeworks/blob/fe-03/tasks/L19>