



НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

# Грешки от контролните

- HTML: Hyper Text Markup Language
- ul, li
- линкове, картинки
- уеб сървърът е по-скоро back end



# Web forms

- Използваме web форми (формуляри), когато е необходимо потребителят да въведе/предостави определена информация в нашият уебсайт
- За да добавим web-форма към нашият html, използваме таг-а `<form></form>`
- `form` е отделен композитен елемент, който съдържа набор от елементи, наречени “полета” на web форма-та
- Елементите на една web форма, могат да бъдат текстови полета (такива в които се въвежда текст), падащи менюта, радио-бутони (тези кръгличките), checkboxes, бутони и скрити полета

## Пример за веб форма:

**Get in touch**

Your name:

email:

Your message:



```
<form action="/api/send-feedback.php">
  <p>
    <label for="sender-name">Your name:</label>
    <input type="text" name="sender-name">
  </p>
  <p>
    <label for="email">email:</label>
    <input type="text" name="email">
  </p>
  <p>
    <label for="content">Your message:</label>
  </p>
  <p>
    <textarea name="content" rows="5"></textarea>
  </p>
  <p>
    <input type="submit" name="submit" value="submit">
  </p>
</form>
```

# Пример за веб форма:

**Get in touch**

**text input** → Your name:

**text input** → email:

Your message:

**text area** →



- `form` елементът също така има атрибути, които определят къде ще отиде информацията, която въвежда от потребителя и как
  - `action="api/send-feedback.php"`
  - `method="post"`
- Кокато потребителят “събмитне” (изпрати) форма-та, информацията, която той е въвел, се изпраща към адресът, указан в `action` атрибута (най-често това е веб сървърът, на който е нашият сайт)
- Съответният веб сървър трябва да знае как да “обработи” заявката и да върне съответното съдържание, отговарящо на резултата от обработката



# GET vs POST

- Two commonly used methods for a request-response between a client and server are: GET and POST.
- GET - Requests data from a specified resource
- POST - Submits data to be processed to a specified resource
- [http://www.w3schools.com/tags/ref\\_httpmethods.asp](http://www.w3schools.com/tags/ref_httpmethods.asp)

# Input полета

- Всичко, което потребителят въвежда на нашата страница, се нарича “input”
- Те могат да бъдат текстови, бутони, списъци с опции или падащи менюта



# Полета за въвеждане на текст

- `<input type="text">`
- ползваме ги, когато искаме юзерът да въведе името си или пък имейл
- в HTML5, email е дори отделен тип инпут (`<input type="email">`)
- input елемент-а няма затварящ таг!

- Полета за пароли
  - `<input type="password">`
  - съдържанието на това поле не се визуализира (вместо това излизат точки)
- Textarea
  - Използваме това поле за въвеждане на по-голям обем текст. Например при съобщение във форма за контакт.
  - `<textarea></textarea>`
  - има затварящ таг!



# Бутони

- Бутони за събмитване
  - `<button type="submit"></button>`
  - `<input type="submit">`
    - button има затварящ таг, докато input няма!
- За ресет-ване на форма
  - `<input type="reset">`
  - изтрива всичко попълнено във формата

# Бутони

- Други
  - Това са бутони, които не събмитват форма-та
  - `<input type="button">`
  - `<button></button>`
- Две основни разлики между `input` и `button`:
  - `input` може да се ползва само в контекста на `form`, а `button` - в целия сайт
  - `input` не може да има "съдържание", защото няма затварящ таг, докато `button` може



# СПИСЪЦИ

drop down  
(select box)



Select a country:

---

Spoken languages:

checkboxes



☐ English

☐ Bulgarian

☐ French

---

Do you like my page?

radio



☐ Yes ☐ No

# Списък от опции без ограничение на избора

- `<input type="checkbox">`
- визуализира се като квадратче и служи за отбелязване на избор от рода на:  
  
[x] Приемам условията за ползване
- ако искаме да предоставим избор от повече опции, можем да групираме няколко такива бутона, чрез атрибута `name`



# Списък от опции с възможност за за максимум един избор

- `<input type="radio">`
- визуализира се като кръгче и има смисъл от него само ако има поне още един такъв, който да е в същата група от радио бутони
- за да групираме няколко радио бутона, трябва да използваме атрибута **name**
- Ако искаме да използваме само 1 радио бутон, значи ни трябва **checkbox**

# Падащи менюта

```
<select>
```

```
<option>Option 1</option>
```

```
<option>Option 2</option>
```

```
...
```

```
</select>
```



- Ползва се ако имаме избор от прекалено много опции (повече от 5)
- може да бъде със single или multiple choice
- единствения форм-елемент, който е система от два елемента, вложени един в друг

# name атрибутът

- Чрез този атрибут задаваме името на параметъра, който ще се изпрати при събмит на форма-та
- Той трябва да е уникален, освен ако не се използва за група от еднотипни полета като checkboxes или radio buttons



# value атрибутът

- Чрез този атрибут задаваме стойността на параметъра, който ще се изпрати при събмит на форма-та
- Използва се при hidden полетата или когато искаме да зададем стойност по подразбиране

# hidden полета

- `<input type="hidden" name="key" value="123426536">`
- Използват се за изпращане на данни, които не се въвеждат директно от потребителя
- Често се генерират от back-end логиката
- Понякога се генерират и от front-end-a



# Други

- Файл ъплоад
  - `<input type="file">`
- HTML5 полета

# Други специални инпут атрибути

- `placeholder="Enter some text here"`
- `readonly`
- `disabled`
- `required`
- `checked & selected`



# Въпроси?

# Домашно

<https://github.com/zzeni/swift-academy-homeworks/tree/master/tasks/L6>