

Стандарти в уеб дизайна и основни уеб технологии

За да създадем работещ уеб сайт, трябва да разбираме как браузърът получава съдържанието и какви технологии го изграждат. Урокът обяснява ролята на уеб стандартите и разграничението между структура, оформление и програмна логика.

1 Организация на работата в уеб пространството

1.1 Хипертекст и хипервръзка

Хипертекстът представя информация чрез свързани документи. Част от текста или друг елемент, например изображение, може да служи като връзка към друга информация. Самата връзка се нарича **хипервръзка (hyperlink)**.

Хипервръзките позволяват да преминаваме към друга страница, файл или част от същата страница. Хипертекстът е основа на **World Wide Web (WWW)** — система от свързани уеб ресурси, достъпни чрез интернет.

1.2 Уеб страница и уеб сайт

- **Уеб страницата** е документ, който браузърът визуализира. Може да включва текст, изображения, звук, видео и интерактивни елементи. Структурата ѝ обикновено се описва с HTML.
- **Уеб сайтът** е съвкупност от свързани страници и ресурси с общо предназначение. Те обикновено са достъпни под общо домейн име. Файловете могат да бъдат организирани в папки и подпапки.

Интернет и WWW не са едно и също: интернет е мрежовата инфраструктура, а WWW е една от услугите, които я използват.

1.3 URL адрес

URL (Uniform Resource Locator) е адрес, който посочва местоположението и начина за достъп до ресурс. Ресурсът може да бъде страница, изображение или друг файл.

Пример: <https://www.example.com/pages/about.html>

- **https** — схемата, указваща използването на HTTPS;
- **www.example.com** — името на хоста;
- **/pages/about.html** — пътят до ресурса.

Домейнът example.com е част от името на хоста. Един и същ ресурс понякога може да бъде достъпен чрез повече от един URL адрес.

1.4 Архитектура клиент сървър

Уеб приложенията обикновено използват **архитектура клиент–сървър**. Клиентът и сървърът са програми с различни роли, които могат да работят на различни компютри.

Клиентът, обикновено браузърът, изпраща заявка и показва получения отговор. **Уеб сървърът** приема заявки и предоставя ресурси. Компютърът, на който работи тази програма, също често се нарича сървър.

Сървърът може да върне готов файл или да използва сървърна програма, която генерира съдържание. Когато е необходимо, тази програма извлича информация от база данни.

Последователност на работа:

1. Потребителят въвежда адрес или избира хипервръзка.
2. Браузърът изпраща заявка към сървъра.
3. Сървърът обработва заявката и намира или генерира необходимото съдържание.

4. Сървърът изпраща отговор към браузъра.

5. Браузърът обработва отговора и визуализира страницата. Може да изпрати допълнителни заявки за изображения, стилове и други ресурси.

1.5 HTTP и HTTPS

- **HTTP (Hypertext Transfer Protocol)** е протокол за обмен на заявки и отговори между клиент и сървър. Чрез него се пренасят HTML документи и други ресурси, включително изображения и файлове със стилове.
- **HTTPS** е HTTP през защитена връзка, която използва **TLS**. Данните се криптират при предаването им, а протоколът осигурява и защита срещу незабелязана промяна на обменяните данни.

Цифровият сертификат помага на браузъра да удостовери сървъра. Често се използва названието „SSL сертификат“, но съвременните защитени връзки използват TLS.

Важно: HTTPS защитава връзката. То не гарантира, че съдържанието на сайта е вярно или че сайтът е защитен от всички видове атаки.

2 Уеб стандарти и тяхното предназначение

Уеб стандартите са спецификации и правила за технологиите, чрез които се създава, обменя и представя съдържание в уеб. Те задават обща основа за работата на разработчици, браузъри и други системи.

При ранното развитие на уеб различни браузъри поддържат собствени разширения. Това създава проблеми със съвместимостта и налага изработването на общи правила. Стандартите се развиват заедно с технологиите.

2.1 Ползи от спазването на стандартите

- **Съвместимост** — по-предвидима работа в различни браузъри и устройства.
- **Качество и поддръжка** — по-ясна структура, по-лесни проверки, поправки и актуализации.
- **Достъпност** — основа за използване на съдържанието от по-широк кръг хора, включително чрез помощни технологии.
- **Ефективност** — по-добре организираният код може да намали времето и разходите за разработване и да подпомогне бързото зареждане.
- **Индексиране** — смислената структура помага на търсачките да разбират съдържанието.

Спазването на стандартите подпомага тези резултати, но не ги гарантира само по себе си. Сайтът трябва да се тества в реални браузъри и на различни устройства.

2.2 Стандартизиращи организации

W3C (World Wide Web Consortium) е международна организация за развитие на уеб стандартите, основана от Тим Бърнърс-Лий. Работата по стандартите включва сътрудничество между организации, специалисти и разработчици.

Организация и адрес	Основна роля и примери
W3C w3.org	Уеб спецификации, например CSS, SVG и стандарти за достъпност; също XHTML, XForms, RDF, GRDDL и OWL.
WHATWG whatwg.org	Поддържа HTML Living Standard и DOM. Работи в сътрудничество с W3C.
Unicode Consortium unicode.org	Развива Unicode — общ стандарт за представяне на символи от различни писмености, използван и чрез UTF-8.
Ecma International ecma-international.org	Разработва стандарти за информационни технологии, включително ECMAScript — основата на JavaScript.
ISO iso.org	Разработва международни стандарти в много области, включително информационни технологии и потребителски интерфейси.
IANA iana.org	Координира DNS кореновата зона, IP адресните ресурси и регистрите на параметри на интернет протоколите.

2.3 Дизайн стандарти и адаптивен дизайн

Дизайн стандартът на организация или проект може да задава шрифтове, цветови комбинации, вид на бутоните, навигация и разположение на търсачката. Така страниците имат последователен вид и поведение. Няма един задължителен шрифт или едно разположение на менюто за всички сайтове.

Адаптивният (responsive) дизайн позволява оформлението да се приспособява към наличната ширина на екрана. Колони могат да се подредят една под друга, изображенията да променят размера си, а менюто да се представи по друг начин. Целта е съдържанието да остане удобно за използване на компютър, таблет и телефон.

3 Езици за създаване на уеб съдържание

Езиците могат да се разглеждат според основната им роля: описание на структурата, описание на оформлението или изпълнение на програмна логика.

3.1 Маркиращи езици

Маркиращите езици използват означения, които описват структурата и смисъла на части от документа. Те могат да обозначават заглавия, параграфи, данни и други елементи.

- **HTML (HyperText Markup Language)** описва структурата на уеб страницата. Чрез него се определят заглавия, параграфи, списъци, връзки, изображения, таблици и формуляри. HTML е маркиращ, а не програмен език.

HTML използва **тагове**, записани между < и >, за обозначаване на елементи. Например в <p>Текст</p> има отварящ таг, съдържание и затварящ таг. Някои елементи, например img, нямат затварящ таг.

```
<h1>Училищни новини</h1>
```

```
<p>Добре дошли в нашия сайт.</p>
```

- **XML (eXtensible Markup Language)** описва структурирани данни. Авторът определя таговете според предназначението на данните. XML се използва за съхранение и обмен на информация между системи.

```
<product>
```

```
  <name>Тетрадка</name>
```

```
</product>
```

Таговете product и name са избрани от автора.

- **XHTML (eXtensible HyperText Markup Language)** представя HTML според правилата на XML. Той изисква строг синтаксис: правилно влагане, затваряне на елементите и стойности на атрибутите в кавички. Имената на елементите и атрибутите в XHTML се записват с малки букви.
- **LaTeX** се използва за подготовка на документи, особено научни текстове със сложни формули, символи и таблици.

3.2 Езици за описание на стилове

Тези езици определят визуалното представяне на съдържанието.

- **CSS (Cascading Style Sheets)** задава шрифтове, цветове, фон, размери, отстояния и подредба. Един набор от стилове може да се използва за много страници.

```
p {  
  color: blue;  
  font-size: 16px;  
}
```

Примерът задава син цвят и размер 16 пиксела на параграфите.

- **DSSSL (Document Style Semantics and Specification Language)** е стандарт и език за обработка и оформление на SGML документи. За нашите уеб страници ще използваме CSS.

3.3 Езици за уеб програмиране

Програмните езици добавят логика: обработване на действия, проверка на данни, изчисления и управление на съдържание. Част от програмите се изпълняват в браузъра, а други — на сървъра.

- **JavaScript** се използва за интерактивни уеб страници. В браузъра може да реагира на натискане на бутон, да проверява въведени данни и да променя съдържание. JavaScript може да се изпълнява и на сървъра чрез подходяща среда.
- **PHP (PHP: Hypertext Preprocessor)** е език за сървърно програмиране. PHP кодът се изпълнява на сървъра, а полученият резултат се изпраща към браузъра. Посетителят обикновено не получава самия PHP код. PHP се използва за динамични страници, формуляри и работа с бази данни.
- **Python** е език с общо предназначение, използван и за сървърни уеб приложения. Позволява обработка на данни и разработване на приложения с различни функции.

- **Java** е език за приложения, включително сървърни системи. Програмите обикновено се изпълняват чрез Java виртуална машина, което подпомага използването им на различни платформи. Java и JavaScript са различни езици.
- **SQL (Structured Query Language)** е специализиран език за работа с релационни бази данни. Използва се за създаване на структури, добавяне, промяна, изтриване и извличане на данни. SQL не определя външния вид на страницата.
- **CGI (Common Gateway Interface)** не е програмен език. Това е интерфейс, който позволява на уеб сървър да взаимодейства с външни програми, например за обработка на формуляри. Самите програми могат да са написани на различни езици.

4 Други технологии в уеб разработването

Уеб технологиите включват не само езици, а и библиотеки, платформи, среди за изпълнение, системи за управление на бази данни и системи за управление на съдържание.

Технология	Предназначение
React	JavaScript библиотека за изграждане на потребителски интерфейси от компоненти.
MySQL и MariaDB	Системи за управление на релационни бази данни, използвани от много уеб приложения.
MongoDB	Система за управление на документна база данни, която съхранява данни като документи.
ASP.NET	Платформа за създаване на уеб приложения и услуги с .NET, например чрез C#.
Node.js	Среда за изпълнение на JavaScript извън браузъра, включително на сървъра.
CMS	Система за управление на съдържание, която улеснява създаването и обновяването на сайт. Пример е WordPress.

5 Избор на технологии за конкретен сайт

Изборът се прави още при планирането и проектирането. Той зависи от темата, целите, потребителите, изискванията на клиента, обема на информацията, необходимите функции и честотата на актуализиране.

- **Малък информационен сайт:** Ако представя няколко страници, без профили и обработка на поръчки, HTML и CSS могат да бъдат достатъчни. JavaScript се добавя при необходимост от интерактивно поведение.
- **Онлайн магазин:** За каталози, профили и поръчки обикновено са нужни клиентска част, сървърна логика и база данни. Възможно е използване на готова платформа или CMS с подходящи разширения. Плащанията могат да се обработват чрез външна платежна услуга.

5.1 Статични и динамични сайтове

- **Статичният сайт** предоставя предварително създадени файлове. За промяна на основното съдържание файловете се редактират или се генерират отново. Примери са малък информационен сайт или портфолио.
- **Динамичният сайт** генерира съдържание според заявката, наличните данни или потребителя. Често използва база данни. Примери са онлайн магазин, социална мрежа и новинарски сайт със система за управление на съдържанието. Динамичният сайт може да показва персонализирано съдържание, например профил или препоръки. Статичен сайт също може да има интерактивни елементи чрез JavaScript. Разграничението е в начина на предоставяне на съдържанието, а не само в наличието на движение или бутони.

5.2 Основни етапи на разработването

1. **Планиране** — определяне на цели, потребители, съдържание и функции.
2. **Проектиране** — структура, навигация, оформление и избор на технологии.
3. **Разработка** — изграждане на страниците и необходимата програмна логика.
4. **Тестване** — проверка на връзки, функции, достъпност и работа на различни устройства.
5. **Публикуване и поддръжка** — предоставяне на сайта онлайн, обновяване и поправяне на проблеми.

5.3 Практическа задача

Отворете сайта на W3C: <https://www.w3.org/standards/>

Изберете две области, например CSS и достъпност. Запишете какво уреждат стандартите в тях и по един пример как спазването им може да подобри уеб сайт. За всяка област са достатъчни два–три кратки реда; не е необходимо да изучавате цялата спецификация.