

# ИЗБРАНИ СОФТУЕРНИ ТЕХНОЛОГИИ ЗА ИЗГРАЖДАНЕ НА ИНТЕРАКТИВЕН ИНТЕРФЕЙС

Детелина Сергеева Милкотева

ПУ „Паисий Хилендарски“ – гр. Пловдив

**Резюме:** Настоящата разработка има за цел да представи най-удачните съвременни софтуерни технологии за изграждане на динамична информационна презентация по атрактивен начин с **интерактивен уеб подход**, наречена **интерактивен интерфейс**.

Една от най-важните задачи е разработването на навигационната система, която ориентира и води посетителите през нея. Напътства създаването на хубави графики или други визуални елементи и за избора на стилове и шрифтове, за направата на уеб сайта привлекателен, атрактивен – **интерактивен**.

**Ключови думи:** *Интерактивен интерфейс, интерактивен уеб подход, интерактивност, интерактивна област, интерактивна презентация, интерактивно приложение*

**„Чудеса се случват, но човек доста трябва да се потруди за това.“**

Трудно може да си представим съвременния свят без информацията, която получаваме от уебсайтовете. Използваме ги за професионални или домашни цели, за лични нужди, развлечения, за лесна комуникация и т. н. Всеки сайт е съвкупност от страници, по които може да се „скача“. Страниците съдържат различни елементи: текст, изображения, видео, анимация, звук, списъци, таблици, форми, хипервръзки. Много често те се описват на HTML (Hyper Text Markup Language – език за маркиране на хипертекст), който непрекъснато може да се надгражда с други съвременни софтуерни технологии.

Основен инструмент за привличане и задържане на вниманието на потребителя е използването и „вграждането“ на **интерактивност в web**. Тъй като самата дума **интерактивност** се използва в много различни ситуации и за обозначаване на множество

различни неща, е необходимо да определим какво ще се разбира под интерактивност в web design, а именно: **Интерактивна ще бъде наричана тази web страница, която реагира по определен начин на действията на потребителя.**

**Основна характеристика в работата и изграждането на *интерактивния интерфейс* е балансът** – добре прецененият компромис между изискванията и възможностите. Може би най-важният момент при изграждането на сайта са тестовете, проведени най-малко на два различни браузъра.

## **1. ОСНОВНИ ПРИНЦИПИ ПРИ ИЗГРАЖДАНЕТО НА ИНТЕРАКТИВЕН ИНТЕРФЕЙС**

**Принципи на уебдизайна. Проектиране на уебсайт. Отзивчив уебдизайн (responsive web design) и адаптивен уебдизайн - същност и спецификации.**

1. Потребителят трябва да се чувства сигурен, докато разглежда сайта, така че във всеки момент да знае къде точно се намира и на къде да продължи.
2. Интерфейсът, чрез който потребителите се придвижват из сайта, често се нарича изглед и усещане (look and feel) на сайта.
3. За да има успех, дизайнът на вашия сайт трябва да бъде преносим и достъпен за потребител, ползващи различни по вид браузъри, операционни системи и компютърни платформи. Анализирайте аудиторията и си изградете профил на типичния потребител.
4. Планирайте страницата си така, че да бъдат достъпни при различна скорост на връзката. Ако страниците ви се зареждат бавно, защото съдържат големи изображения или сложни анимации, потребителите ви ще напуснат сайта още преди да видят съдържанието му.
5. Целта Ви е да организирате и представите информацията по логичен и лесен за навигация начин. Трябва да предоставите на потребителите си разнообразни навигационни средства, без да ги отклонявате от крайната им цел. Вместо да създавате дълги, превъртащи се страници, разделете информацията на по-малки сегменти и ги свържете посредством хипертекст.

6. Планирайте единен облик за всички секции и страници на вашия сайт. Подсилете елементите, придаващи индивидуалност, и създайте плавни преходи от една страница към друга чрез повтарящи се цветове и шрифтове и чрез оформление на страниците, което позволява йерархични нива. Избягвайте неочакваните резки промени, освен ако това не е целта ви.  
**!!! Нека когато минават от една уебстраница към друга, потребителите все едно разлистват списание. Въпреки че всяка страница трябва да бъде завършена изцяло, тя същевременно е и част от целия сайт. Дизайнът на отделната страница трябва да отразява идентичността на сайта като цяло, независимо къде е позиционирана в йерархията на сайта. Пример: [ucd.bg](http://ucd.bg) и с клик върху банера, препраща към началната страница.**
7. Решетъчната структура е метод за оформление, при който елементите по страницата се организират в редове и колони. Можете да използвате решетъчна структура, за да постигнете визуално единство на всички страници в сайта.
8. Важно е обаче да се съобразите и с начина, по който потребителят взаимодейства със съдържанието – усещането от сайта!
9. Как да направим унифициран дизайн за нашия Уеб?: Когато създавате уебсайт, планирайте обща тема и структура за всички страници в него. Изборът на цветове, шрифтове, изображения и оформление трябва да формира единна визия, която да улесни потребителите на сайта ви. Темата трябва да съответства на впечатлението, което искате да създадете за вашата организация. Изборът на меки цветове, четливи шрифтове и отделно структурирани колони подчертава двете основни теми – информационна и новинарска. Когато създавате дизайна на сайт, не трябва да мислите за всяка страница поотделно. За да бъде сайтът ви хармоничен и унифициран, предвидете плавни преходи, базирайте визуалната структура върху решетка и използвайте активни бели полета.
10. Помислете как потребителят иска да взаимодейства с информацията на вашата уебстраница. Изградете дизайна според типа на съдържанието и вземете под внимание дали посетителят ще чете, или просто ще преглежда страниците ви.

11. Разположението на елементите също е много важно: Трудно е да се предвиди в какъв ред потребителят ще прегледа страницата. Все пак, съществува общоприето схващане за относителната тежест на отделните области от екрана.

2		
5	1	3
4		

- Най-важните зони са номерирани по възходящ ред. Най-много окото гледа в центъра на уеб страницата, за това там се поместват най-важните акценти.
12. С помощта на хипертекста речниците и други документи с висока информационна плътност стават много по-лесни за ползване.
13. Възможно е да претрупате дадена страница с твърде много информация. Осигурете подходяща навигация, за да бъде локализирано бързо търсеното съдържание. Разделете съдържанието на малки порции и използвайте хипертекстови връзки, за да го разпределите между страниците.
14. Не забравяйте хората с увреждания! – Всеки сайт, ориентиран към широката публика, трябва да очаква, че потребители с физически увреждания също ще искат да разгледат съдържанието му. Осигуряването на достъпност при дизайна на даден сайт означава страниците да бъдат конструирани по такъв начин, че да са достъпни независимо от физически, сензорни или когнитивни увреждания, работни ограничения или технологични пречки от страна на потребителя.
15. **Отзивчив уеб дизайн (Responsive Web Design** или **RWD**; „*responsive*“ от английски – отзивчив), представлява вид стилизиране на уеб страници, целящо да осигури оптималната ѝ визуализация и улеснено взаимодействие с нея, което да е подходящо за различни устройства (мобилни телефони, планшети, настолни компютри и др.) и техните различни резолюции на екрана.

При адаптивния уеб дизайн са налични няколко оформления на сайта с определени фиксирани размери за различните устройства:

- 320, 360 пиксела за смартфони;
- 768 – за планшети;
- 960 – за компютри.

## 2. ОСНОВНИ СТИЛОВИ РЕШЕНИЯ И ТЕХНОЛОГИИ

Много важни са цветовете усещания. На фиг. 1 са дадени основните цветове.

**Фиг. 1** Имена на цветове, познати за повечето браузъри (цветовете с получен шрифт са част от уеб палитрата)

Име на цвят	Шестнадесетична стойност (Hex)	Име на цвят	Шестнадесетична стойност (Hex)
Aqua	00FFFF	Navy	000080
Black	000000	Olive	808000
Blue	0000FF	Purple	800080
Fuchsia	FF00FF	Red	FF0000
Grey	808080	Silver	C0C0C0
Green	008000	Teal	008080
Lime	00FF00	White	FFFFFF
Maroon	800000	Yellow	FFFF00

**Контрол на цвета чрез CSS: Стойност на цвят /свойство color/** - позволява да определите цвета на всеки елемент от уебстраницата. То едновременно задава цвета на текста и рамката на елемента, освен ако отделно не сте посочили цвят на рамката чрез нейните свойства:

P {color: blue;} /\*име на цвят\*/

P {color: #0000FF;} /\*шестнадесетична стойност\*/

P {color: rgb (0, 0, 255);} /\*rgb числа\*/

P {color: rgb (0%с 0%, 100%)} /\*rgb процент\*/

**Стандартен цвят на текста** – Цветовете се наследяват от родителските към дъщерните елементи. Ако посочите цвят за <body>, всички елементи на страницата ще наследят неговия цвят, което е ефективен начин за избор на стандартен цвят на текста за целия сайт. Следното правило определя цвета на елемента <body>: body {color: #006633;}.

**Промяна цвета на връзките** – можете да променяте цвета на хипервръзки, като използвате следните специални класове от CSS: link – цвят на не посетена връзка, active – цвят на активна връзка, visited – цвят на посочена връзка/по подразбиране е лилав/.

**Цвят на фона /свойство background-color/:** позволява да задаваме цвят за фон на всеки елемент от страницата. Това включва и всяко вътрешно бяло поле. Може да се прилага както при елементи от тип блок, така и при съдържателните елементи. Следното стилово правило селектира елемент span и му задава цвят на фона: span {background-color: #ffffff;}  
Правилото е приложено към елемента span, който огражда името на автора в кода:  
<span>Detelina Milkoteva</span> - съдържателен елемент с цвят на фона.

**Цвят на фона на страницата:** за да изберете фонов цвят на цялата страница, използвайте body селектор. Така цветът ще се отнася за цялата съдържателна част. По подразбиране цветът на всеки елемент е прозрачен. Затова всички елементи показват фона на самата страница, освен ако отделно не им е зададен друг цвят. Следното правило посочва цвят за фона на елемента <body>: body {background-color: #cccccc;}.

**Обръщане цветовете на текст и фон:** смяна на цвета на текста (обикновено черен) и фона (обикновено бял) е често срещан ефект в печатните издания. В Интернет това може да се постигне с избрани от вас произволни цветове. Този похват най-често се прилага за заглавия не толкова при стандартен текст. Осъществява се лесно чрез стилово правило. Следното правило прави цвета на фона на елемент <h1> син, а цвета на текста – бял. H1 {color:#ffffff; background-color: blue; padding: .25 em;}.

**Цвят на фона на таблици:** лесно се променя цвета на фона на таблици с различен цвят, но винаги като използвате свойството background-color. То може да се прилага към всички елементи: таблица <table>, ред <tr>, заглавна клетка <th>, <td> клетка за данни.  
table {background-color: #ffffff;} /\*фонът на таблицата е бял\*/  
tr.hilite {background-color: yellow;} /\*фонът на редовете от клас "hilite" са жълти\*/  
th.header {background-color:##cc9966;} /\*фонът на клетките от клас "header" е светлокафяв\*/

**Добавяне на елемент <img>: непосредствено след отварящия таг <p>:**

```
<html>  
<head>
```

```
<title>заглавие</title>
</head>
<body>
<h1>.....</h1>
<p>/*може да се опише картинката*/
</p> </body> </html/>
```

**Контрол на фонови изображения с CSS:** Поставяне на фоново изображение /background-image/ - позволява да определим изображението, което е избрано за фоново. В стандартния HTML единствената възможност е изображението да се наслагва многократно, докато запълни целия прозорец на брауъра. Това е и стандартното поведение на свойството background-image в CSS.

**Стандартни шрифтове:** шрифтовете са слаба страна на HTML, защото информацията за тях се съдържа при потребителя. Може единствено да разгледаме как изглежда сайтът ни през различни платформи. За това ползвайте цели семейства шрифтове, за предпочитане без серифни: Arial, Helvetica, Times New Roman.

#### **Общо свойство за шрифт:**

Стойност: {<font-style> || <font-variant> || <font-weight>}?

<font-size>[/<line-height>?<font-family>]

База: виж отделни свойства

Отнася се за: всички елементи

Наследено: да

Проценти: позволени само при <font-size> и <line-height>

#### **Отстъп на абзац – свойството text-indent**

p {text-indent: 2em;}

blockquote {text-indent: - 2em;}

**Свойство text-align – подравняване:** служи за избор на хоризонтално подравняване на текста в даден елемент – left, right, justify, center. P {text-align: justify;}.

**Свойство Декорация / text-decoration/** - подчертаване на текст, ефект, който има особено значение за хипертекстови документи. Това свойство предлага стойност blink, което прави текста да мига. Ако искаме да махнем подчертаването от елементите <a>, то:

```
a {text-decoration: none;}
```

Уебсайтовете, ориентирани към потребителя, са проектирани да отговарят на нуждите на целевата аудитория. Интуитивната, лесна за използване навигационна система е основен елемент. Дизайнът на ориентираните към потребителя уебсайтове има следните характеристики:

- Проектиран за конкретна целева аудитория.
- Фокусира се върху съдържанието, а не върху технологията.
- Предлага лесна за използване навигационна система.
- Организира съдържанието чрез колони и други разделители.
- Събира информация чрез лесни за използване формуляри.
- Използва CSS (каскадни стилкови таблици).

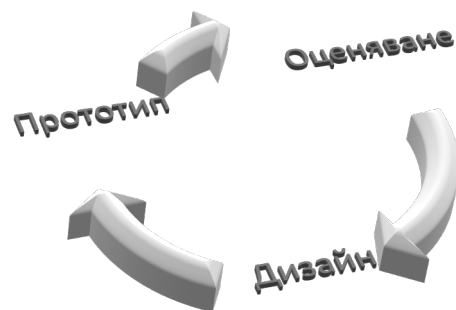


**Фиг. 2 Уебдизайн, ориентиран към потребителя**

Планирането на сайта винаги се извършва преди каквато и да е разработка. С помощта на потребителя сайтът може да бъде изграден така, че да включва функции, които са най-важни за клиентите.

Проучването на желанията на потребителя и прилагането на идеите в дизайна на сайта ще повишат удовлетвореността на потребителите.

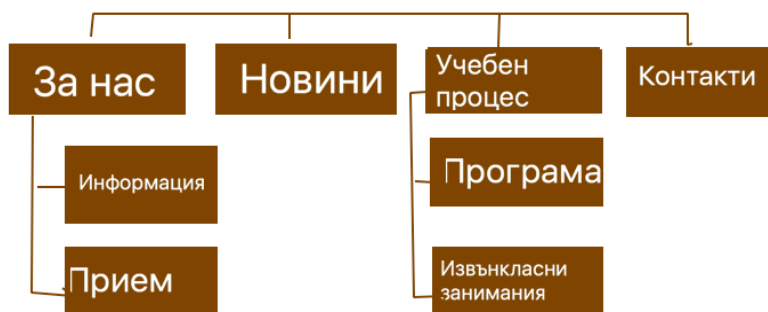




**Фиг. 3 Процес на итеративен дизайн**

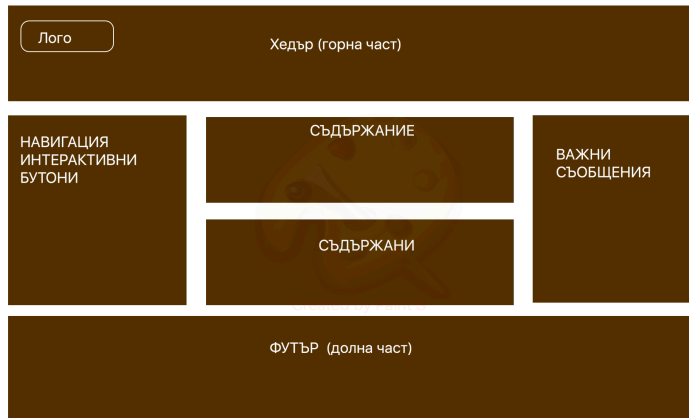
Ако има съществуващ уебсайт, процесът на итеративен дизайн може да бъде полезен в актуализиране на информацията и поддържане на дизайна в крак с технологичните стандарти.

**Фиг. 4 Примерна карта на сайт от уебсайт на едно училище**



Имайте предвид, че картите на сайт не заместват механизмите, базирани на обхождане (crawl-based mechanisms), които търсачките използват, за да открият URL адреси.

**Фиг. 5 Примерна схема на началната страница (Wirefram)**



Вижте снимки на схеми на страница, създадени от професионален веб дизайнер, на адрес <http://deeplinking.net/paper-web/>. Може да видите как изглежда и сайтът [ucd.bg](http://ucd.bg).

### 3. ДРУГИ ТЕХНИКИ И ТЕХНОЛОГИИ

**Стилът.** Важното е, изберем ли стил, да го следваме и да го запазим консистентен в целия сайт. Стилът се отнася както до визуалните елементи и тяхната цялост, така и до текстовете, съдържанието на сайта.

**Снимките.** Каталози или мултимедийни представяния, и как те ще се адаптират за сайта.

**Анимираните елементи.** Понякога има залитане по отношение на анимираните елементи или цялостните Flash решения. Важно е да се знае, че човешкото възприятие има определени граници, и ако искаме да фокусираме вниманието на посетителя на сайта върху конкретни послания, е добре да анимираме или тях, или да не анимираме прекалено много елементи в сайта. В този смисъл – движение на сайта- да, но умерено и с мисъл.

**Простотата.** „Далай Лама веднъж казал, че простотата е ключът към щастието в модерния свят.“ Тази философия може да се приспособи към областта на веб дизайна и *дизайна на дигитални интерфейси*“.

Така започва статията „Направи го просто, тъпако!“ на Пар Амквист (цялата статия е на адрес [www.groovemanifesto.net](http://www.groovemanifesto.net)). Простотата е изключително важна в много аспекти на днешния живот, сам по себе си претоварен от комуникации и информация. Но всички ще се съгласят, че постигането на простотата е трудно. В

този смисъл дизайнът не бива да се натрапва и да доминира, а следва да изведе в правилна емоционална посока посетителите и да позволи на цялото съдържание, и най-вече на важното от него да изпъкне по адекватен начин.

## FLASH

### **Употреба на Flash в уеб страниците:**

Анимации на страници

Интерактивни навигационни ленти

Анимирани рекламни плакати

Интерактивни и мащабируеми карти

Интерактивни форми

Цели интерфейси на уеб сайтове, които да заместят традиционните HTML страници

Анимационни филмчета

Музикални видео филми

„Джубокс“ – интерфейс за изпълнение на MP3 файлове

Голяма част от аспектите на файловия формат Flash го правят подходящ за добавяне на интерактивно съдържание към уеб страници:

- Малък размер на файловете.
- Мащабируем е.
- Високо качество на изображенията.
- Използва поточна технология. Flash файловете започват да се изпълняват бързо.
- Използва вграден звук. Flash е един добър начин за предоставяне на фонов звук и насочени към потребителя звукови ефекти на уеб сайта.
- Flash е добре поддържан формат.
- Поддръжка на скриптове. Flash използва езика за писане на скриптове ActionScript за контролиране поведението на съдържанието.
- Той е свободно достъпен формат. Macromedia направи файловия формат Flash общодостъпен, което означава, че другите разработчици могат да вграждат поддръжка на Flash в своите приложения. Едно от първите такива

приложения е LiveMotion на Adobe (инструмент за създаване на **интерактивни бутони** и анимирани обекти), което съхранява файловете си във Flash формат, за да могат в последствие да бъдат изпълнени от възпроизвеждащото устройство на Flash.

### *Мощни Flash инструменти*

Можете да създавате доста модерни конфигурации с Flash направо в прозореца на програмата посредством вградените му характеристики. За голяма част от потребителите това е абсолютно достатъчно. Но Flash не се използва само за анимация и звукови ефекти. С помощта на усъвършенстваните скриптові характеристики или допълнителните софтуери, Flash филмите могат да бъдат програмирани така, че да изпълняват **интерактивни функции** и дори да служат като край за динамичното генерирано съдържание.

### **PHP – ВГРАЖДАНЕ В HTML**

Първото, с което трябва да се запознаете, е взаимовръзката между PHP и HTML.

```
<HTML>
```

```
<HEAD>Sample PHP Scrip</HEAD>
```

```
<BODY>
```

The following prints “Hello, World”:

```
<?php print “Hello, World”;
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

От примера се вижда, че PHP кодът е вграден в HTML кода. Когато интерпретаторът на PHP достигне отварящия таг<?php, той изпълнява кода до маркера?>. След това PHP заменя този PHP код с резултата от изпълнението му (ако има такъв), а останалият текст

(HTML текста) се пропуска непроменен до уеб клиента. Така споменатият скрипт ще даде следния резултат:

```
<HTML>
<HEAD>Sample PHP Script</HEAD><BODY>
The following prints "Hello, World":
Hello, World
</BODY>
</HTML>
```

**PHP** се вгражда лесно в HTML файлове, показващи HTML формуляри за събиране на данни от потребителите и файлове, качвани от тях.

След като определите нужните части за Вашето **интерактивно уеб приложение**, трябва да ги организирате в едно цяло. Един от често използваните методи за организиране се нарича „един скрипт за всичко“, което означава, че index.php обработва всички заявки за различни страници.

**JAVA APPLETС** – Java е обектно-ориентиран програмен език, разработен от Sun Microsystems . Трябва да се отбележи, че той не е свързан с JavaScript, който е език за скриптове, разработен от Netscape Navigator, и който работи в HTML документите в брауъра. Тъй като Java е пълен програмен език /подобно на C, C++/, той може да се използва за създаването на цялостни приложения. Първоначалния принос на Java към Мрежата обаче, беше под формата на Java аплети – самосъдържащи се, малки изпълними програми. Тези програми, именувани с наставката class, могат да се поставят направо в уеб страницата, подобно на графики.

#### **Предимства и недостатъци на Java аpletите:**

Аpletите са идеални за уеб разпространение поради следните причини:

- Те не зависят от платформата;
- Те се свалят и изпълняват изцяло на клиента, така че сървърът не се натоварва продължително;

- Обикновено файловете на аpletите са доста компактни и бързо се свалят от сървъра;
- Те не изискват специален патентен plug-in за инсталирането си. Сега всички основни браузъри поддържат Java, което означава, че шансовете потребителите да видят вашия аplet са доста големи.

Разбира се, всяка утопична технология си има своя тъмна страна и за съжаление в реалния свят браузърите могат да проявят темпераментно отношение по отношение на начините, по които манипулират Java applets. Браузърите са прословути със своите сринове в присъствието на жадните за изчисления аплети. Най-общо казано, на браузърите им отнема много време да инициализират Java, което означава че прогонва потребителите.

Вероятно съществуват стотици Java applets. Това е просто най-общ списък на нещата, които те правят:

- Помощни програми – калкулатори, календари, часовници, електронни таблици;
- Текстови ефекти – превъртащи се подвижни текстове, пулсиращ текст, блестящи цветни текстови съобщения;
- Аудио ефекти – дигитални „китари”, радиобутони;
- Игри – Asteroid, Hangman, Minesweeper, кръстословици;
- Разни – диаграми на биоритми, симулатори на битки, ежедневни котировки и др.

Едно добро начало може да се окаже разделът за аплети в сайта на Java на Sun на адрес: <http://java.sun.com/applets/>. Тази страница предоставя списък от връзки към източници за аплети.

Ако търсите полезни аплети, които да можете да ползвате на минутата, проверете в JavaBoutique на адрес: <http://javaboutique.internet.com/>. Тук ще намерите стотици налични аплети, придружени с ясни инструкции за употребата им. Този сайт предоставя чудесна възможност да добавите **интерактивни ефекти** на страниците си, без да учите програмиране.

**Добавяне на аплети в страница:** По настоящем съществуват два начина за добавяне на аплети към веб страница: с инструкцията за форматиране <object>, която се препоръчва от HTML 4.01, и с по-добре поддържаната инструкция за форматиране <applet>. Ето и двата метода:

#### *Добавяне на аплети с <object>:*

Може да добавите елементарен самосъдържащ се аplet към HTML документ с помощта на инструкцията за форматиране <object>, ето така:

```
<OBJECT CLASSID="applet.class"CODEBASE="http://somedomain.com/classes/">
```

An applet with some useful function should display in this space.

```
</OBJECT>.
```

#### *Добавяне на аплети с <applet>:*

Инструкцията за форматиране <applet> представлява контейнер за някакъв брой параметрични инструкции за форматиране (<param>). Ето как би изглеждала инструкцията за форматиране <applet> за някаква игра:

```
<APPLET CODEBASE=class CODE="Wacky.class" WIDTH=300 HEIGHT=400>
```

```
<PARAM NAME="Delay" VALUE="250">
```

```
<PARAM NAME="Time"VALUE="120">
```

```
<PARAM NAME="playSounds" VALUE="YES">
```

```
</APPLET>
```

## **МИГРАЦИЯ ОТ HTML КЪМ XHTML**

Кога да изберем XHTML пред HTML?

Ако създавате нов сайт, може да използвате строго форматирани XHTML, CSS за оформлението му. Ако вече имате сайт, Ви предстои повече работа . Миграцията на работещ сайт към XHTML трябва да бъде постепенен процес, а не рязък преход към нещо ново. XHTML позволява да започнете да прилагате новия синтаксис, докато старият HTML все още контролира някои атрибути, като например цветовете на страницата и връзките. Мигрирайки и спазвайки строго правилата в XHTML, все повече ще изчиствате кода на

съществуващите страници, ще планирате структурата на кода за новите страници, ще премахвате отпадналите вече елементи и ще отделяте оформлението в отделен CSS файл. Ето и стъпките, през които трябва да преминете при миграция от HTML към XHTML:

1. *Оценка на съществуващия код*
2. *Оценка на съществуващото оформление*
3. *Използване на CSS*
4. *Съвместимост със старите версии*

За да бъдете ефективни, трябва да умеете свободно да работите на ниво HTML код.

В крайна сметка, докато стигнете до крайни вариант на страницата си, най-вероятно ще сте работили с комбинация от няколко различни софтуерни инструмента, за да придадете **интерактивност на вашия уеб интерфейс**.

#### **Създаване на качествен код**

В миналото уеб дизайнерите се базираха на принципа „важното е да работи“ и пишеха нестандартен код, с който да накарат брауъра да изобрази желанието от тях резултат.

Илюстрация за това е използването на заглавните тагове от <h1> до <h6> изцяло заради размера на буквите, а не според логическото им предназначение в даден документ. Друг пример е използването на <table> като инструмент за позициониране на обекти вместо като контейнер за данни, каквато е действителното му предназначение.

Подобни лоши навици, за щастие, вече са в миналото. Необходимо е да създавате код, който да е максимално съобразен със стандартите и да представя съдържанието на уеб сайта по най-добрия и **интерактивен уеб подход**.

- Придържане към стандартите
- Използване на семантично маркиране
- Избор на правилния тип документ
- Валидиране на кода



## **КРАТКО ВЪВЕДЕНИЕ В XML**

XML — разширяем маркиращ език (PME)) е стандарт (метаезик), дефиниращ правила за създаване на специализирани маркиращи езици, както и синтаксисът, на който тези езици трябва да се подчиняват. Сам по себе си той е безполезен, защото указва само как да бъде структуриран един документ (чрез маркиране с етикети), но не и какво означават отделните маркери (етикети). Ето защо е по-правилно да се нарече метаезик отколкото език — обикновените езици имат семантика, т.е. предават някаква информация, докато XML указва само граматиката (по-точно синтаксиса) на езиците, базирани на него.

Тези програмни средства могат след това да се използват и в изработването на софтуер който е написан за конкретен XML език. Или накратко — разделянето на синтаксиса от семантиката е поредното въплъщение на принципа „Разделяй и владей“.

**JAVA SCRIPT** е главния клиентски скриптов език, който е вграден в повечето браузъри и може да изпълни всякакъв вид функции в рамките на уеб страницата, без да има нужда от помощта на сървъра. Използва се за манипулиране на изображенията и текста, за стартиране на малки изкачащи прозорци, за разширяване и свиване на падащи менюта, за събиране на информация от формуляри и като цяло за динамични елементи на сайтовете. Скриптовете, изпълнявани от сървърната страна, свързват уеб страницата с база от данни или приложение, позволяващи да се настрои сайта за различни потребители и да се обслужват динамични страници (онлайн магазини, онлайн разплащания и други).

Други технологии са XHTML, DHTML, ASP.NET MVC 5, а за индивидуален дизайн на банер, лого, бутон или друг елемент от уеб страниците, се използват CorelDraw, Photo Shop, Illustrator. А най-мощните средства за създаване на динамични, „интелигентни“ **или интерактивни уебстраници са СИСТЕМИТЕ ЗА УПРАВЛЕНИЕ НА СЪДЪРЖАНИЕ.** Функционалността е тази, която различава уеб сайта от брошурата, презентацията, мултимедийния диск. Тя прави сайта интерактивен. Различни функционални модули и елементи дават възможност потребителят на сайта (посетител, клиент или администратор) да взаимодейства с неговото съдържание и да изпълни конкретна задача.

Системата за управление на съдържанието (Content Management System), наричана още админ панел или контрол панел, е задължителен компонент на всеки

съвременен сайт. Той представлява вътрешната, защитена (вход с потребителско име и парола) част от сайта.

Добрият админ панел дава възможност на собственика на сайта сам да актуализира съдържанието на сайта и да управлява голяма част от неговите функционалности. Така необходимостта от намеса от страна на разработчика възниква доста по-рядко и обикновено във връзка с добавяне или корекции на техническо ниво.

Ето основните неща, които можете да искате да се изпълняват чрез админ панела на вашия сайт:

- Редакция и форматиране на текстове;
- Автоматично оразмеряване на снимките и илюстрациите (при дефинирани точни размери и по колко варианта ще има всяка една снимка);
- Добавяне на нови страници и категории;

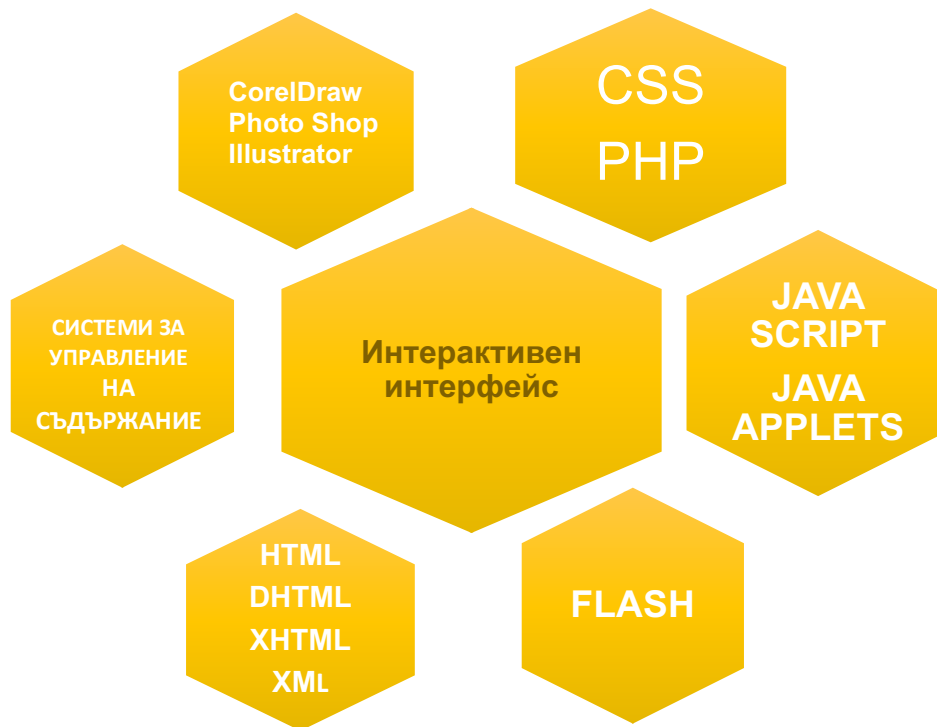
Всички останали функционалности в сайта, част от които ще разгледаме в тази глава, също се интегрират в админ панела.

Обикновено веб агенциите използват собствена разработка за CMS, която се адаптира за нуждите на конкретния проект. Освен това е възможно използването на готови CMS решения, обикновено софтуер с отворен код, като Joomla, Mam boo, Word press, OS Commerce (за е – магазин специално) и др.

**Техническа поддръжка.** Разбира се, сайтът е софтуерно изграден и техническата поддръжка в повечето случаи се свежда до редовно подновяване на домейна и хостинга на сайта, както и актуализирането му.

### **Обратната връзка.**

Обратната връзка разкрива една от възможностите за **интерактивност**, които предоставя Интернет. Нейната употреба прави сайта „жив“, показва отношение към посетителя и би могло да е от полза при негово доразработване.



## ЛИТЕРАТУРА:

**Жустин Томс, Венцислав Джамбазов** Основи на WEB дизайна, изд. Сиела-Софт ЕНД Паблицинг София 2004; ISBN 954-649-694-4;

**Джоел Склар** Принципи на уебдизайна– трето издание; изд. Дуо Дизайн, ISBN 978-954-8396-34-9;

**Йен Дехаан** Macromedia FLASH MX 2004 – официален учебен курс; изд. СофтПрес ООД 2004, ISBN 954-685-296-1;

**Дори Смит** Java за World Wide Web; изд. Peachpit Press, ISBN 0-201-35340-7;

**Сергей Соколов** CSS 3 в примери, изд. Асеновци 2009; ISBN 978-954-8898-13-3;

# SELECTED SOFTWARE TECHNOLOGIES FOR BUILDING INTERACTIVE INTERFACE

**Detelina Sergeeva Milkoteva**  
**University of Plovdiv "Paisii Hilendarski"**

**Summary:** The present development aims to present the most up-to-date software technologies for building a dynamic information presentation in an attractive way with an interactive web approach called an interactive interface. One of the most important tasks is to develop a navigation system that guides and guides visitors through it. Guides the creation of nice graphics or other visual elements and the choice of styles and fonts, to make the website attractive, attractive – interactive.

**Keywords:** *Interactive interface, interactive web approach, interactivity, interactive area, interactive presentation, interactive application*

Detelina Milkoteva, tel.: 0882 42 47 01; [detelina\\_milkoteva@abv.bg](mailto:detelina_milkoteva@abv.bg)