

pawel@cs.uni.wroc.pl

<http://pawel.ii.uni.wroc.pl/>

Mobile first

Plan wykładu

- Wprowadzenie
- Znaczniki
- Walidatory
- Best practices
- Co zrobić dla mobilków?
- Responsive Web Design
- Kilka technikaliów
- Domena

Wprowadzenie

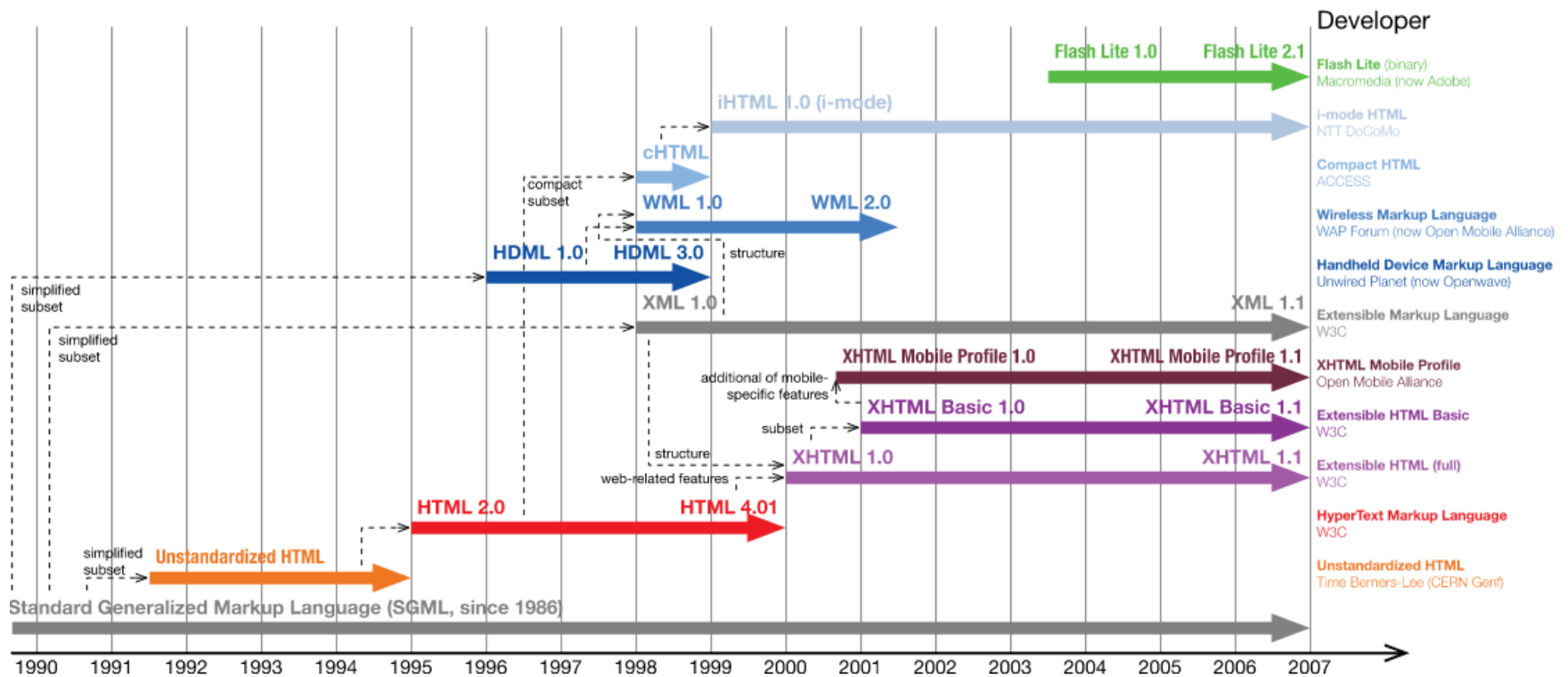
- Po co tworzyć strony mobilne?
 - Na rynku jest coraz więcej urządzeń mobilnych
 - Ruch mobilny jeśli już nie jest, to wkrótce będzie większy niż ten generowany z PC
- Kto jest „targetem” takich stron?
 - Użytkownicy szukają stron usług, sklepów w „okolicy”
 - Wyszukiwanie produktów i cen
 - Coraz częściej przeglądanie Internetu na tabletach
- Stan obecny
 - wolne łącza vs. ładna, szybko ładowana strona
- Obecnie celem jest utworzenie serwisu działającego na: telefonie, tablecie, laptopie, desktopie, itd.

Znaczniki

- Starsze:
 - WML
http://en.wikipedia.org/wiki/Wireless_Markup_Language
 - cHTML
<http://en.wikipedia.org/wiki/C-HTML>
- Nowsze
 - XHTML Basic
http://en.wikipedia.org/wiki/XHTML_Basic
 - XHTML Mobile Profile
http://en.wikipedia.org/wiki/XHTML_Mobile_Profile
http://technical.openmobilealliance.org/technical/release_program/docs/browsing/v2_3-20080331-a/oma-ts-xhtmlmp-v1_2-20080331-a.pdf
 - HTML5

Znaczniki

Evolution of Mobile Web-Related Markup Languages



Źródło: wikipedia

Walidatory

- Kilka ważniejszych
 - <http://validator.w3.org/mobile/>
 - <http://ready.mobi/>
 - <http://validadores.tawdis.net/mobileok>
 - <http://wurml.shadowplay.net/sect.php?sectid=5>

Best practices

- **Szybkie ładowanie stron**
 - Użytkownicy zwykle są w ruchu i szybko potrzebują informacji
 - Strona powinna być nastawiona na informację podaną zwięźle
 - np. wypunktowania zamiast dużych bloków tekstu

Best practices

- **W stronach nastawionych na interakcję**
 - Formularze proste jak się da: mało pól, mało kroków do dokonania transakcji
 - Jak się da, zamiast pól używaj list rozwijanych, checkboxów itp.
 - pamiętaj, że dane trzeba wprowadzić klawiaturą ekranową
 - Wprowadzaj numery telefonów tak, żeby po kliknięciu można było zadzwonić

Best practices

- **Łatwa nawigacja**
 - Szczególnie ważne z uwagi na mały ekran
 - Staraj się unikać przewijania
 - szczególnie w obu kierunkach, pionowe jest ok
 - Do 7 linków nawigacji per strona
 - Stała dostępność pola wyszukiwania na złożonych stronach

Best practices

- **Lokalne otoczenie**
 - Łatwy dostęp do lokalnych usług i sklepów
 - Wyszukiwarka po adresach
 - Wykorzystanie map
 - Wykorzystanie GPS, jeśli możliwe

Best practices

- **Przyjazny dla obsługi palcem**
 - Duże przyciski z marginesami
 - Linki, checkboxy, itp. łatwe do trafienia (z padding)
- **Ciągłość interakcji**
 - Zapamiętywanie wyszukiwań
 - Dostęp do skrótów na wszystkich stronach

Best practices

- **Czytelny design**

- Przyciski i treści powinny kontrastować z tłem
- Treści powinny być czytelne bez powiększania
- Warto dodawać efekty 3D, cienie pod przyciski i inne ulepszenia

Best practices

- **Przekierowania dla urządzeń mobilnych**
 - Po wejściu na stronę serwisu, przekieruj do wersji mobilnej
 - Ale daj możliwość powrotu do wersji desktop
 - Zapamiętaj, na której wersji serwisu użytkownik został
 - Daj możliwość wyboru wersji strony
 - Kod do pobrania ze strony
<http://detectmobilebrowsers.com/>
- Powyższe rozpatrujemy, gdy tworzony jest dedykowany serwis na urządzenia mobilne

Best practices

■ Dostępność serwisu

- Nie używaj Flasha
- Do animacji i bajerów używaj HTML5
- Pamiętaj, że smartfon może być pozycji poziomej i pionowej

■ Dobry serwis

- Zapytaj użytkowników, czego oczekują od wersji mobilnej
- Używaj analiz, aby wiedzieć, co jest używane w serwisie
- Stale ulepszaj 😊

Co zrobić dla mobilków?

- Nic
 - nie ma kosztów ani dodatkowej pracy
 - ale tracimy użytkowników korzystających z urządzeń mobilnych
- Dedykowana osobna strona mobilna
 - Wiąże się z utworzeniem nowej strony
 - Użytkownicy mają wersję mobilną, ale...
 - Pojawia się problem synchronizacji treści pomiędzy wersjami
 - ale jest to dobre podejście, gdy serwis mobilny oferuje inny zakres treści niż strona stacjonarna
 - Trzeba dobrze zrobić mechanizm wyboru wersji i ewentualnego przełączania się między nimi (użytkownik może sobie zepsuć i nie umieć naprawić)

Co zrobić dla mobilków?

- Wersja generowana na serwerze
 - Wiąże się z utrzymywaniem dwóch wersji kodu
 - Nadal trzeba zrobić dobrze detekcję urządzenia, w którym użytkownik wyświetla stronę
- Responsive Web Design
 - Jedna wersja strony
 - Celem jest utworzenie strony na wszystkie urządzenia: telefon, tablet, laptop, desktop

Terminologia wokół Responsive Web Design

- Fixed vs. Adaptive vs. Responsive vs. Fluid Web Design
 - Fixed
 - Sztywny, nie reagujący na zmianę rozdzielczości
 - Fluid
 - Struktura szablon jest oparta o procenty i zmienia się płynnie wraz ze zmianą rozdzielczości urządzenia
 - Adaptive
 - Tworzymy kilka layoutów dla kilku wybranych zakresów rozdzielczości, np.
 - Np. width < 480, < 768, >768
 - Np. width > 1024, >768, >480, <480
 - Dla łączącego się urządzenia wybierany jest jeden z przygotowanych layoutów
 - Responsive
 - Tworzymy kilka layoutów skupiając się na optymalnej prezentacji treści
 - Nadal będzie kilka break pointów, ale niekoniecznie powiązane z rozdzielczościami urządzeń
 - W ramach jednego zakresu rozdzielczości layout powinien być „fluid”

Responsive Web Design

- **Podejście Mobile First**
 - Polega na utworzeniu szablonu dla smartfonu
 - Wraz ze wzrostem rozdzielczości, dokładamy kolejne lub przebudujemy istniejące elementy strony
 - **Główne zalety**
 - Skupiamy się na istotnych elementach treści
 - Dzięki startu od mobile, w wersji desktop też często dodajemy elementy użyte w mobile (np. lokalizacji)

Mobile Layout Design

- Główne elementy szablonu (nie tylko) mobilnego
 - Logo
 - zwykle na górze po lewej
 - Nawigacja
 - zwykle na górze po prawej
 - można także rozważyć tab bar (powszechnie używany w iOS) jako element nawigacji
 - Search
 - zwykle pole tekstowe jest ukryte, ale dostępna jest ikonka, której kliknięcie wyzwała pojawienie się pola tekstowego
 - Treść
 - najważniejsza część strony

Mobile Layout Design

- Content area
 - Trzeba przemyśleć, jak pokazać treść w mobile i desktop, przykładowo:
 - Dla tekstu:
 - Pojedynczy box przy większej rozdzielczości może pokazać więcej (reszta jest ukryta)
 - Może być przycisk „show more”
 - Dla galerii:
 - Pojedynczy obrazek dla mobile (z przyciskami nawigacyjnymi)
 - Układ tabelaryczny dla desktop

Mobile Layout Design

- Navigation
 - Kilka sposobów na realizację menu w mobile:
 - Tak jak w desktop – np. horizontal menu jak będzie się zmniejszać rozdzielczość, to będzie się zawijać
 - Combo box
 - Pop down, ewentualnie pop left
 - Ikonka, której kliknięcie wysuwa menu
 - Ostatnio modny sposób

Responsive Web Design

- Viewport
 - Sam termin określa obszar przeglądarki, w którym prezentowana jest strona.
 - Poprzez tag `<meta name="viewport" content="...">` określamy wygląd wyświetlania stron w urządzeniu mobilny
 - `<meta name="viewport" content="width=device-width">`

Responsive Web Design

- Media Query
 - Główna nowość CSS3 wykorzystywana w RWD
 - Określa media type (np. screen) oraz wyrażenie (np. dot. rozdzielczości ekranu). Jeśli wyrażenie jest prawdziwe, skojarzony styl jest pobierany
 - Problemem jest brak wsparcie dla IE < 9, ale wtedy z pomocą przychodzą biblioteki
 - Media types to (te najczęściej spotykane)
 - screen, print, all

Responsive Web Design

- Media Query: wyrażenia
 - Wyrażenie tworzymy:
 - Feature: value, gdzie value jest opcjonalne (np. portrait)
 - Większość własności można poprzedzić min i max
 - Lista wszystkich
 - width, height, device-width, device-height, orientation, aspect-ratio, device-aspect-ratio, color, color-index, monochrome, resolution, scan, grid
 - ... ale zwykle używa się tylko kilku wybranych

Responsive Web Design

- Media Query: najczęściej używane features
 - width
 - `<link rel="stylesheet" media="screen and (min-width: 800px) and (min-height: 1024px)" href="normal.css">`
 - orientation
 - `<link rel="stylesheet" media=" all and (orientation:landscape)" href="website.css">`
 - `<link rel="stylesheet" media=" all and (orientation:portrait)" href="website.css">`

Responsive Web Design

- Media Queries: jak używać?
 - W linku do pliku CSS
 - `<link rel="stylesheet" media="..." href="..." />`
 - Po regule import
 - `@import('plik.css') screen and (max-width:800px)`
 - Wykorzystując regułę @media
 - `@media screen and (max-width:800px) { ... }`
 - W kodzie JavaScript
 - `if (window.styleMedia.matchMedium("screen and (max-width:800px)") { ... }`

Responsive Web Design

- Jeszcze raz o formularzach
 - Etykiety kontrolek
 - Nad kontrolką
 - Standardowe podejście w desktop, czyli etykieta po lewej raczej się nie sprawdzi
 - Użycie tzw. placeholders, czyli etykieta jest w samej kontrolki i znika, po ustawieniu „focus-u”
 - Jest atrybut placeholder w HTML5, są też odpowiednie biblioteki w JS
 - Ważne, aby łatwo odróżnić opis kontrolki od wprowadzonego tekstu
 - Używamy kontrolek różnych typów z HTML5
 - Pierwszy zysk: pokażą się kontrolki (np. dla daty), gdzie nie trzeba używać klawiatury, tylko wybrać wartość
 - Spełniony jest postulat wspomniany już wcześniej
 - Drugi zysk: w przypadku, gdy klawiatura jest jednak niezbędna, zostanie ona dostosowana do wprowadzenia wartości, czyli np. numeryczna dla liczb, czy też zostanie dodany znak @ w przypadku e-maila

Responsive Web Design

- Jeszcze raz o formularzach c.d.
 - Raczej unikamy form wizards, wygodniej użytkownikowi będzie przewinąć ekran i wypełnić dalej formularz
 - Wykorzystanie geolokacji do zaproponowania miasta lub kodu pocztowego – mniej pisania dla użytkownika

Kilka technikaliów

- Obrazków bardzo mało, generalnie strona musi być bardzo lekka
 - zalecany rozmiar obrazków do 200x150px
- Wielkość czcionki w wartościach bezwzględnych (najlepiej pt)
- Generalnie staramy się unikać podawania wielkości w pikselach

Kilka technikaliów

- Nie używać
 - map obrazkowych
 - popup-ów
 - auto-odświeżania strony
 - ramek
 - zagnieżdżonych tabel
 - tabeli do budowy struktury
 - skryptów JS, jeśli to możliwe
- Jak najmniej linków do zewnętrznych serwisów
- Używać buforowania
- Strona powinna być czytelna także bez stylów

Domena

- Jeśli decydujemy się na osobny serwis, pojawia się kwestia domeny
- Dla mobilnych jest specjalna domena .mobi
- Ale sugeruję poddomenę m.
 - Jest za darmo
 - Nie ma problemu z dostępnością
 - Łatwiej pozycjonować całość w Google
 - Obecnie to już prawie standard

Literatura

- Zbiór linków

<http://support.google.com/webmasters/bin/answer.py?hl=pl&answer=72462>

- Krótki przewodnik

<https://developers.google.com/webmasters/smartphone-sites/>

- Best practices od W3C

<http://www.w3.org/TR/mobile-bp/>

- Wikipedia

http://en.wikipedia.org/wiki/Mobile_Web

http://en.wikipedia.org/wiki/HTML5_in_mobile_devices

- W3C

<http://www.w3.org/Mobile/>

Literatura

- Responsive Web Design

- Przegląd tematu

- http://en.wikipedia.org/wiki/Responsive_web_design

- <http://msdn.microsoft.com/en-us/magazine/hh653584.aspx>

- Fluid vs. Adaptive vs. Responsive

- <http://teamtreehouse.com/library/build-a-responsive-website/introduction-to-responsive-web-design/fixed-fluid-adaptive-and-responsive-2>

- <http://www.hyperarts.com/blog/what-is-responsive-adaptive-and-fluid-design-defining-terms-in-a-multi-device-world/>

- Viewport

- https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport_meta_tag

- <http://blog.kamilbrenk.pl/meta-tag-viewport-kiedy-i-jak-uzywac/>

- <http://menacingcloud.com/?c=cssViewportOrMetaTag>

- Media Queries

- <http://blog.kamilbrenk.pl/css3-i-media-queries/>

- <http://www.w3.org/TR/css3-mediaqueries/>

Literatura

- Responsive Web Design

- Zbiór przykładów

- <http://mediaqueri.es/>

- 20 fajnych stron:

- <http://www.creativebloq.com/css3/ethan-marcottes-20-favourite-responsive-sites-10112931>

- Zasoby

- <http://bradfrost.github.io/this-is-responsive/resources.html>

- Framework twitter bootstrap

- <http://getbootstrap.com/>

- Media Query Debugging

- <https://github.com/bueltge/Simple-Media-Queries-Debugging>

- Jak zaplanować breakpointy?

- <http://www.metaltoad.com/blog/simple-device-diagram-responsive-design-planning>