



# Plan prezentacji

- rozwiązania, w których Phonegap się nie sprawdza
- natywnie nie znaczy lepiej...
- różnice w renderowaniu między platformami mobilnymi
- biblioteki JS bez spadków wydajności
- co tak naprawdę spowalnia aplikacje www? Mity i fakty
- ręce precz od SVG
- kilka sztuczek praktycznych

# "Programuj wyłącznie natywnie!"

- "Programowanie stron internetowych to nie programowanie"
- "Nie umiesz programować natywnie?!"
- "W czym programujesz? W HTMLu? Ha ha"
- "Programowanie w językach interpretowalnych to nie programowanie"

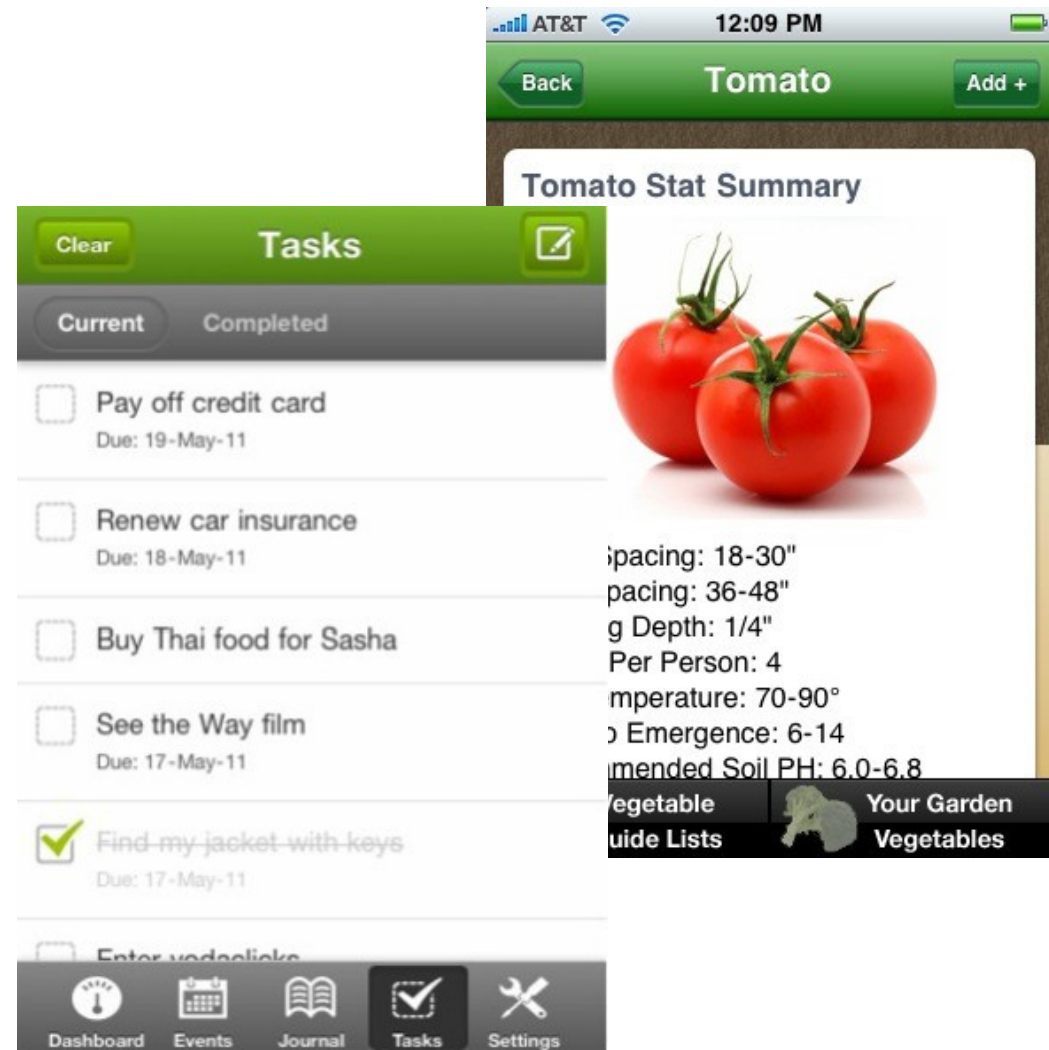
F\*\*\* you \* !

Fight with real arguments, you hater!



# Zastosowanie Phonegapa

- **czytniki HTML**
- komunikatory
- proste gry
- aplikacje użytkowe
- front dla aplikacji serwerowych



# Phonogap - przykłady niewłaściwych zastosowań

- gry zręcznościowe i grafika 3D
- aplikacje narzędziowe
- obsługa multimedialnych
- obróbka dużej ilości danych (np. zdjęć)

# Programowanie w HTML5, CSS3, JS

(na chwilę założymy, że Phonegap to tylko  
pudełeczko do włożenia HTML5)

## HTML



# biblioteki JS

- multibiblioteki (jQuery, mootools)
- frameworki mobilne (jQuery Mobile, Sencha Touch)
- efekty wizualne (cycle, rotat9r)
- mikrobiblioteki do mikrozasosowań (lazyload)

# Frameworki (mobilne)

- jQuery Mobile
- Sencha Touch
- KENDO UI
- JUNIOR (<http://justspamjustin.github.com/junior/>)



# Sencha



Lepiej trzymać się od nich z daleka



**Kendo UI**  
THE ART OF WEB DEVELOPMENT



# jQuery

mobile framework.

# Po co nam jQuery?

- Selektory
- AJAX
- Metody dodatkowe (np. \$.parseJSON(), itd)
- Tworzenie kolejek zdarzeń synchronicznych
- Animacje
- ~~Zmiana wyglądu elementów HTML~~

# "jQuery jest duże i powolne"

- ...waga kilku bibliotek byłaby większa
- nie wymyślajmy koła od nowa
- odchudzona wersja 2.0 (bez hacków IE)
- właściwie stosowane nie powoduje spowolnienia

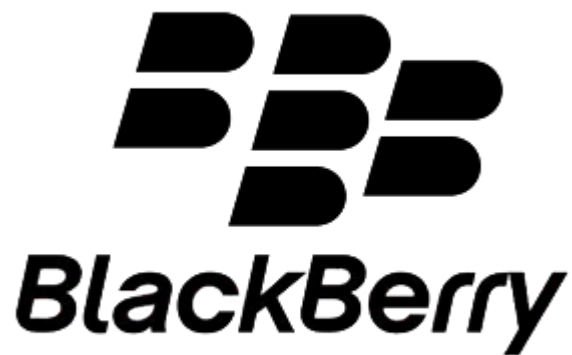
# Zepto JS?

- *If you use jQuery, you already know how to use Zepto* - tylko nie bądźmy zaskoczeni tym, że coś nie działa ;)
- poprawia tak naprawdę tylko animacje
- brak pełnej obsługi wszystkich metod jQuery
- brak pełnej obsługi AJAXa względem JQuery
- Ostatnia zmiana v1.0rc1 - 09 Apr 2012





# Platformy dla Webappek



# iOS

- ideał dla Webappek?
- position:fixed od iOS5 - z pewnymi błędami
- webappki bez "opakowania"
- **brak segmentacji rynku**
  - jedna (w przybliżeniu)
  - słuszna (tutaj można polemizować ;))
  - najnowsza  
wersja systemu



# Android

- średnia wydajność
- wysoka segmentacja rynku
  - $\leq 2.1$  - tego już na szczęście prawie nie ma
  - 2.2 & 2.3 - bardzo duża ilość urządzeń i średnia praca Webbappek
  - 3.0 - ile urządzeń na tym pracuje?
  - 4.0 - duże zmiany w dobrym kierunku
  - 4.1 - kilka poprawek względem 4.0
  - 4.2 - ???



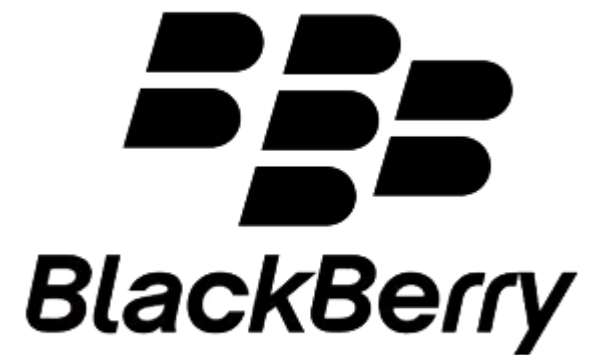
# Windows Phone

- Bardzo słaba wydajność
- Niezgodność ze standardami / przyjętymi konwencjami
- Tylko częściowe otwarcie się na programistów spoza kręgu zaufania MS
- Segmentacja rynku
  - 7.5 / 7.8 (macie nowy ekran startowy i cieszcie się)
  - 8 - tylko najnowsze telefony (pokrycie w granicach błędu statystycznego)
  - Windows RT?



# BlackBerry

nie mnie to oceniać...



Transformacje

Formularze

# Wsparcie HTML5 & CSS3 przez "mobilki"

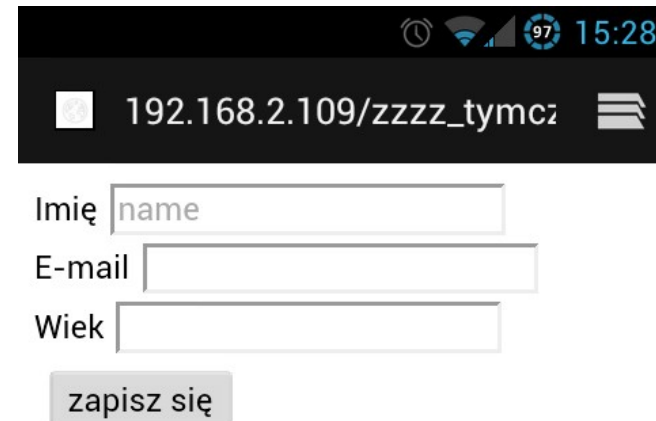
LocalStorage

WebSQL

# Formularze w HTML5

```
<form>
  <label for="username">Imię</label>
  <input id="username" type="text" placeholder="name"><br />
  <label for="email">E-mail</label>
  <input id="email" type="email"/><br />
  <label for="age">Wiek</label>
  <input id="age" type="number"/><br />
  <input type="submit" value="zapisz się" />
</form>
```

# Formularze w HTML5 - Android



192.168.2.109/zzzz\_tymcz

Imię

E-mail

Wiek

- Poprawne działanie w 4.0
- Input "range" nie działa
- Błąd z ostatnim polem number poprawiony w 4.1

# Formularze w HTML5 - iOS



T-Mobile.pl 15:30 39%

← 192.168.2.109/zzzz\_tymcza

Imię

E-mail

Wiek

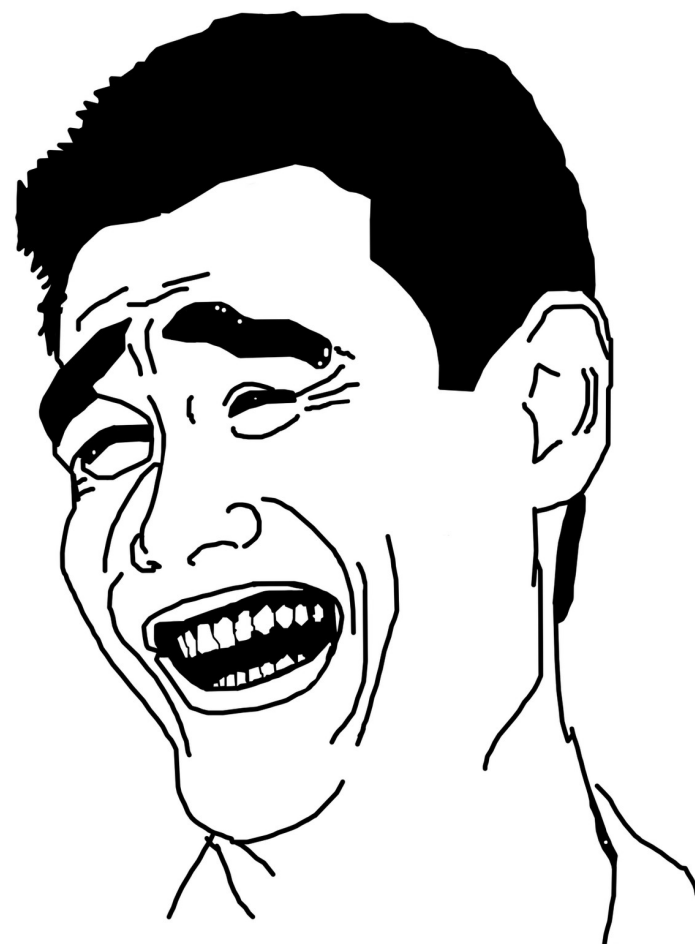
zapisz się

- Pełna obsługa kontrolek (nawet range)



# Formularze w HTML5

## Windows Phone



# Windows Phone

## czyli spacer po polu minowym

(ale za to w kolorowym ubranku)

- TouchStart → MSPointerDown
- prefix -ms-
- CSS filter
- "Dajemy wam ECMA Script 5"



# Windows Phone czyli spacer po polu minowym

## Viewport hack

```
if (navigator.userAgent.match(/IEMobile\/10\.0/)){  
    var msViewportStyle = document.createElement("style");  
    msViewportStyle.appendChild(  
        document.createTextNode(  
            "@-ms-viewport{width:auto!important}"  
        )  
    );  
    document.getElementsByTagName("head")[0].  
    appendChild(msViewportStyle);  
}
```

# Co tak naprawdę spowalnia webappki?



# Co tak naprawdę spowalnia webappki?

- źle zaprojektowana budowa logiczna aplikacji
- źle zaprojektowany layout
- wygląd elementów wymagający używania dodatkowych plików graficznych
- operacje na wyglądzie elementów w JS
- niektóre atrybuty CSS
- skomplikowane reguły CSS (w tym RESETy CSSa)

# Ciężki CSS

- Position:fixed!!!
- Cienie
- Zaokrąglenia
- Tła zawieszane fixed
- Tła cover, contain, size
- @font-face
- Nakładanie przezroczystości
- Gradienty

# Ciężki JS

- Dziesiątki bibliotek doczytywanych na starcie bez względu na to czy są potrzebne
- Dziesiątki aktywnych listenerów dostępnych cały czas
- Animacje przy pomocy JS
- Zewnętrzne statystyki
- Widżety FB i inne śmieci
- Operacje na obiektach SVG

# Ciężki JS

```
var newWidth = aDiv.offsetWidth + 10; // Read  
aDiv.style.width = newWidth + 'px'; // Write  
var newHeight = aDiv.offsetHeight + 10; // Read  
aDiv.style.height = newHeight + 'px'; // Write
```

Nieco lżej...

```
var newWidth = aDiv.offsetWidth + 10; // Read  
var newHeight = aDiv.offsetHeight + 10; // Read  
aDiv.style.width = newWidth + 'px'; // Write  
aDiv.style.height = newHeight + 'px'; // Write
```

# Mity o prędkości JS

- Budowanie obiektów i późniejsze wkładanie ich w DOM
- Przenoszenie obiektów po DOMie
- Systemy templatek / widoków JS

To wszystko prawda, ale nie ma to **ŻADNEGO** zauważalnego znaczenia w porównaniu do:

- błędnej koncepcji aplikacji
- błędnego użycia CSSa
- przeladowania aplikacji grafiką
- użycia animacji JS

# Ale benchmarki mówią co innego...

"Opinie wyrabiane na podstawie syntetycznych testów są mniej / więcej tak sensowne jak nauka jazdy samochodem przez czytanie jego instrukcji obsługi"

Testing in Firefox 21.0 on Ubuntu 64-bit		
	Test	Ops/sec
<b>Chained</b>	<code>\$('#theDiv').addClass('test').removeClass('test');</code>	18,410 ±5.61% fastest
<b>Separate calls</b>	<code>\$('#theDiv').addClass('test');</code> <code>\$('#theDiv').removeClass('test');</code>	12,395 ±7.21% 34% slower
<b>Separate calls (cached)</b>	<code>var d = \$('#theDiv');</code> <code>d.addClass('test');</code> <code>d.removeClass('test');</code>	18,153 ±6.05% fastest

# Czy w takim razie webappki muszą wyglądać tak?!

[Exit](#)

HeadPhones Master

[Learn Practise Time Trial](#)

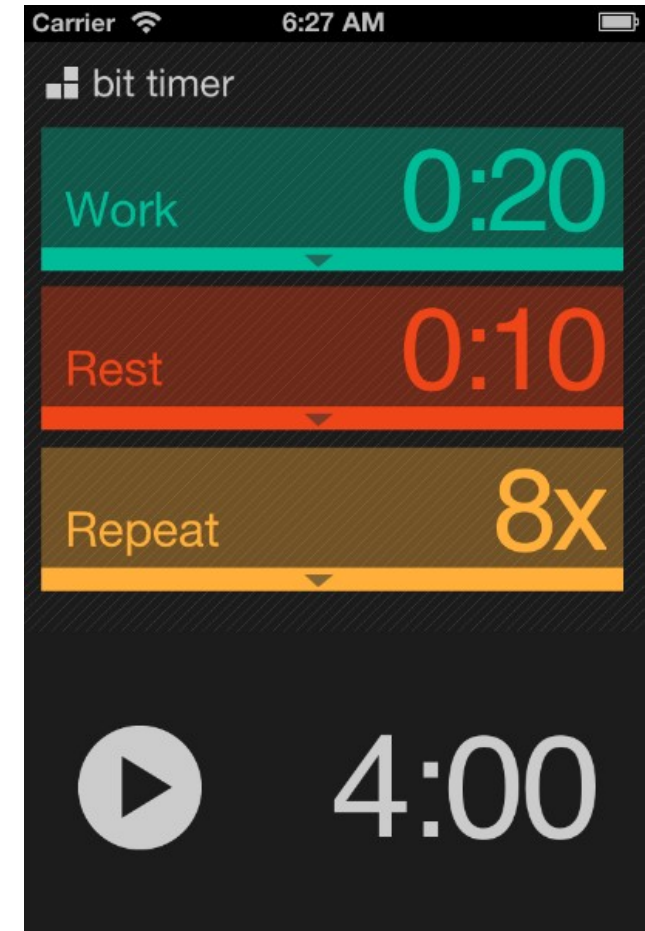
[Options About Exit](#)

[Menu](#)

**Learn**

« »

# Mogą wyglądać tak



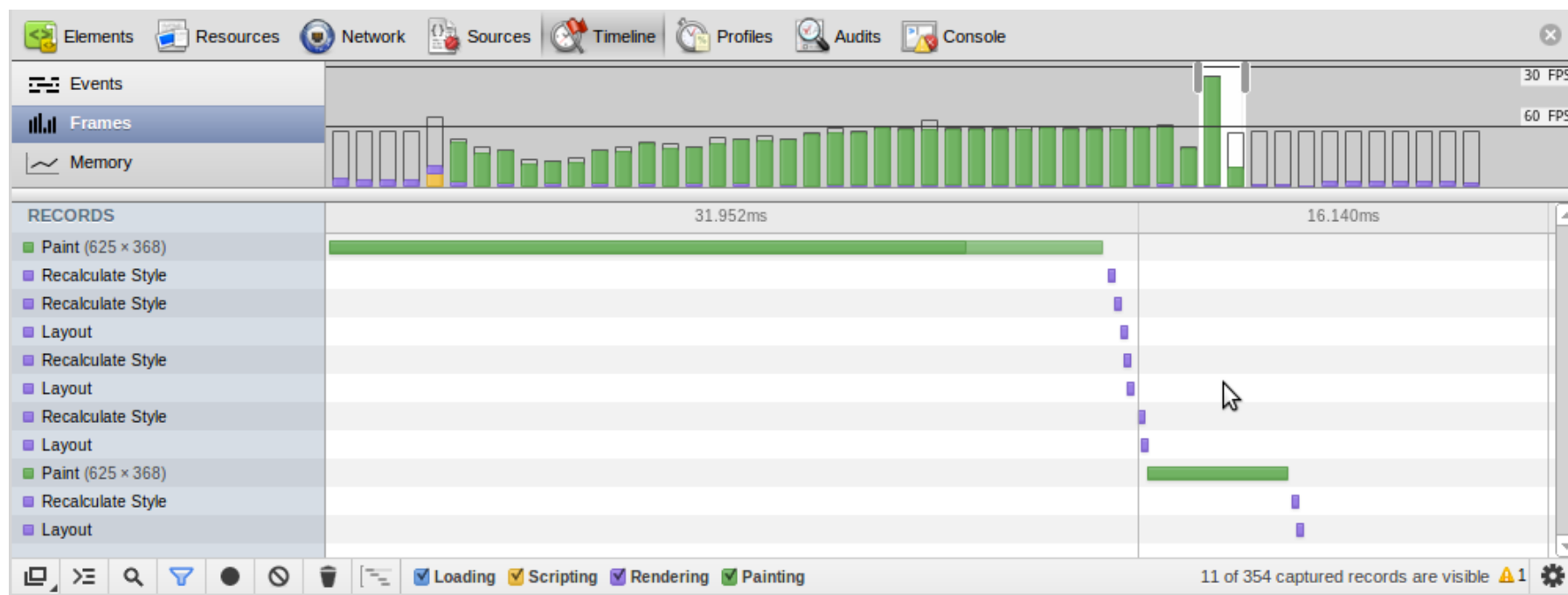
# ~~SVG~~

- Niska wydajność na platformach dekstopowych
- Bardzo niska wydajność na platformach mobilnych
- Przykład: Raphaël + gRaphaël

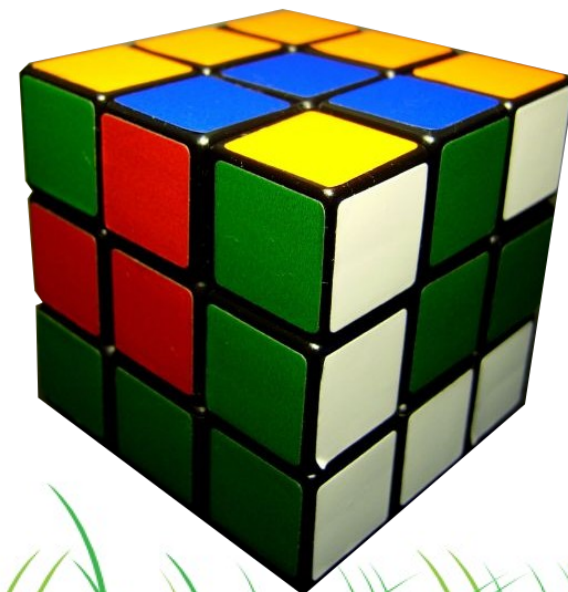


# Kilka zagadnień praktycznych

- "szybsze kliknięcia" - zdarzenie touchstart zamiast click
- repainting - ograniczenie odmalowywania całego ekranu
- "keyframes" - szybsze animacje
- testy wydajności na urządzeniach fizycznych
- audyty szybkości w narzędziach przeglądarek



# Problemy



# Problem 1 - wykresy

Aplikacja wyświetlająca wykresy notowań giełdy surowcowej

## **Rozwiązanie:**

Przetwarzanie danych na serwerze  
i przesyłanie żądanego obrazka.  
SVG nie jest dobrym pomysłem!

# Problem 2 - orientacja ekranu

Wygląd zmieniający się w zależności od orientacji  
i cechujący się inną funkcjonalnością

Aplikacja do przeprojektowania

# Problem 3 - generalna wydajność

"Na HTC G1 / HTC Wildfire / SE Xperia Mini  
działa to trochę za wolno przyspieszcie to..."

*Sam se przyspiesz człowieku... telefon...  
Natywnie działa cokolwiek szybciej?*

# Problem 4 - transformacje grafiki

"(...) i tutaj by taki napis wjeżdżał,  
a potem to okienko robi się okrągłe  
i wtedy odstawiamy ekran startowy"

Z pokorą bierzemy się za OpenGL!  
Webowo tego nie zrobimy...

# Problem 5 - użycie narzuconej z góry biblioteki

Kierownik projektu, który nie jest programistą:  
"Zróbcie to w tym - przecież są biblioteki to musi to działać"

*Ale przecież to będzie mulić...  
Testował ktoś to w ogóle?*

**Rozwiązanie:**

Rozwiążmy problem  
z wydajnością, zanim się pojawi

# Problem 6 – portowanie aplikacji z iOSa na Androida i Windows Phone

To, że to działa teraz szybko na iOSie nie znaczy, że będzie działać szybko na Androidzie.  
Czy będzie w ogóle działać na Windows Phone?

Rozwijajmy aplikację od początku na najwolniejszej platformie!

# Problem 7 - dolne menu

## Kilka podejść:

- position: absolute i przeliczanie JS - duże spowolnienie
- position: fixed - spowolnienie przy przewijaniu
- dwie instancje "przeglądarki"  
i przekazywanie sterowania przez natywny kod
- przewijanie iScroll (słaba wydajność w Androidzie)
- przewijanie górnego zbiornika (nie w Androidzie <= 2.3)
- wyeliminowanie dolnego menu ;)
  - górne menu przyklejone do "strony" (bez fixed)
  - użycie przycisku sprzętowego MENU (ale co z iOSem?)

Należy wybrać rozwiązanie adekwatne do danej aplikacji

# Problem 8 - formularze i position:fixed

Focus na kontrolkach formularzy → position:fixed  
zmienia się na position:absolute

**Rozwiązanie:** Ukrywanie elementów fixed przy focusie



# Problem 9 - renderowanie na platformach innych niż iOS

Przyspieszenie renderowania aplikacji

## **Rozwiązanie:**

- rozpoznanie platformy w JS
- dodanie klasy "ios" do <body>
- reguły CSS rozszerzające efekty graficzne

# Problem 10 - usuwanie stron aplikacji z DOM

Ciągłe usuwanie i dodawanie elementów obniża wydajność  
Rozrost drzewa DOM obniża wydajność

**Rozwiązanie:**  
Sztuka kompromisu...

# Problem 11 - wszystko naraz czy na bieżąco?

Wczytywanie całego drzewa DOM przy starcie?

## **Kilka podejść:**

- Aplikacja składająca się z małej ilości obiektów - wszystko przy starcie
- Rozbudowana aplikacja z dużą ilością widoków (strony) - doczytywanie każdego widoku, kiedy tylko jest potrzebny i kasowanie pozostałych
- Doczytywanie widoków na bieżąco i gromadzenie ich w DOM

# Przydatne narzędzia

- Audit panel z Chrome / Chromium
- Firebug + YSlow
- CSS Explain (<http://josh.github.com/css-explain/>)
- <https://developers.google.com/speed/pagespeed/insights>  
generalnie dla stron, ale...

# Przydatne linki

- [https://developers.google.com/mobile/articles/fast\\_buttons](https://developers.google.com/mobile/articles/fast_buttons)
- [https://developers.google.com/mobile/articles/webapp\\_fixed\\_ui](https://developers.google.com/mobile/articles/webapp_fixed_ui)
- <http://gent.ilcore.com/2011/03/how-not-to-trigger-layout-in-webkit.html>
- <http://www.sencha.com/blog/the-making-of-fastbook-an-html5-love-story>
- <http://www.slideshare.net/AddyOsmani/jquery-proven-performance-tips-tricks>
- <http://jsperf.com/detach-vs-dom-opetarion>

# Informacje o autorze

## Krzysztof "Procek" Ścira

- absolwent AGH  
Informatyka Stosowana  
spec. Informatyka w Hutnictwie
- właściciel i programista  
w desi9n.pl
- miłośnik kolarstwa szosowego,  
boksu i piłki nożnej



Krzysztof  
"Procek"  
Ścira

# Bonus: rotat9r

- wtyczka do jQuery
- alternatywa dla "cycle"
- animacje JS zastąpione transformacjami CSS



Orange vegetable

Orange fruit

rotat9r

...better than "cycle"  
jQuery plugin

Dziękuję :)

rotator

...better than "cycle"  
jQuery plugin

