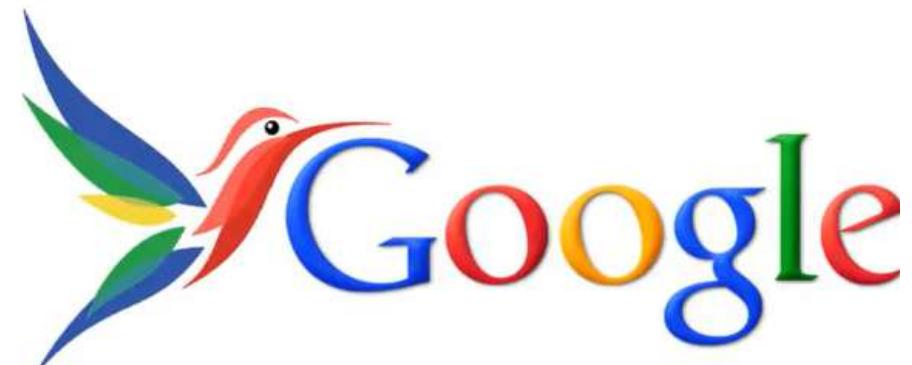


And... LOD?



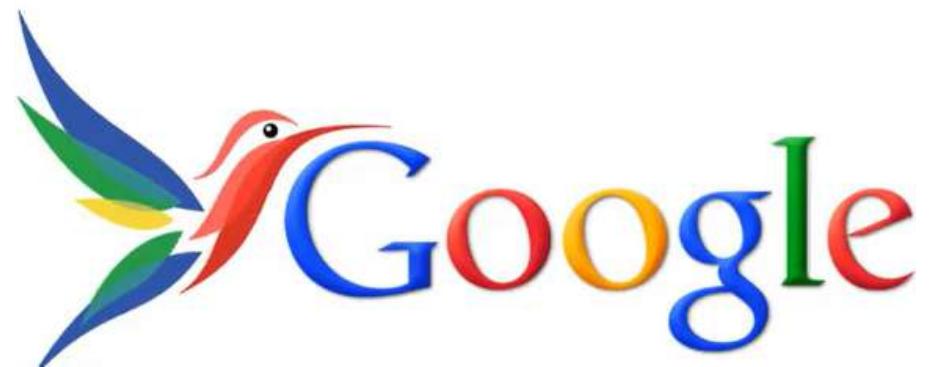
- ◆ LOD = Linked OPEN Data ?
- ◆ Offered in various ways



Example



- ◆ **Freebase**
- ◆ Data offered as ***N-Triple RDF***, also with endpoints





How much data?

- ◆ Size in **2013**: 585 milioni di triple
(circa 47 Gb di dati)
- ◆ Size in **2014**: 2,5 miliardi di triple
(circa 200 Gb di dati) !
- ◆ Size in **2015**: 3,1 miliardi di triple
(circa 250 Gb di dati) !
- ◆ Now...?

<http://www.firebaseio.com/>



404. That's an error.

The requested URL / was not found on this server. That's all we know.



And so...?

◆ Let's see.....(!)





Main Page Discussion

Read View source View history

Search Wikidata



Main page
Community portal
Project chat
Create a new item
Recent changes
Random item
Query Service
Nearby
Help
Donate

Lexicographical data
Create a new Lexeme
Recent changes
Random Lexeme

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Wikidata item

In other projects

Wikimedia Commons
MediaWiki
Meta-Wiki
Multilingual Wikisource
Wikispecies



Welcome!

Wikidata is a free and open knowledge base that can be read and edited by both humans and machines.

Wikidata acts as central storage for the **structured data** of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others.

Wikidata also provides support to many other sites and services beyond just Wikimedia projects! The content of Wikidata is available under a free license, exported using standard formats, and can be interlinked to other open data sets on the linked data web.

Get involved

Learn about data

New to the wonderful world of data? Develop and improve your data literacy through content designed to get you up to speed and feeling comfortable with the fundamentals in no time.



Item: Earth (Q2)



Property: highest point (P610)



custom value: Mount Everest (Q513)

And so...?

- ◆ Let's see.....(!)
- ◆ ... and also....(!!)



Google Search Central > Knowledge Graph Search API

Search

Get Started

Introduction

Prerequisites

How To...

Install Client Libraries

Authorize Requests

Use the Knowledge Graph Search Widget

Terms of Service

Home > Search Central

> Knowledge Graph Search API

Was this helpful?  

Google Knowledge Graph Search API

On this page

Typical use cases

Sample request

Knowledge Graph entities

The Knowledge Graph Search API lets you find entities in the [Google Knowledge Graph](#). The API uses standard [schema.org](#) types and is compliant with the [JSON-LD](#) specification.



Let's now talk...

- ◆ ... of the ***mobile web*** and ***apps***
- ◆ Apps are not «web» but they are anyway related (UI) and a big market



Mobile usability

- ◆ Having seen classic usability and its principles, we can now just focus in the main ***differences***



Premessa

- ◆ Primo aiuto:
Google's mobile compatibility test



Premise...



“Webmaster spam”...





Analysis



- ◆ First, we start from the ***causes*** of the differences between the classic and mobile world: a different ***execution model*** due to the...



Three Base Components of mobile





I

First Component

- ◆ Being ***mobile*** (!)





Second component



- ◆ The ***screen size***, which is obviously quite smaller







Third Component

- ◆ The interaction mean: **fingers** and not the mouse



Note: beware of the target...

- ◆ Facebook
- ◆ Not so well known fact, Facebook has three mobile versions (!!!):
 - ◆ m.facebook.com
 - ◆ touch.facebook.com
 - ◆ 0.facebook.com



The three Facebooks



- ◆ m.facebook.com is what we think we see when we use a smartphone: instead, internally it's touch.facebook.com :
- ◆ m.facebook.com is actually the non-touch version, whereas touch.facebook.com is for touch-based smartphones

And 0.facebook.com?



◆ Back to that soon.

0



Let's see the implications



- ◆ Let's start from the first: being **mobile**
- ◆ Mobile has fundamental consequences
- ◆ The first?



The network!

- ◆ The connection type changes
- ◆ Fact: 3G networks are on 40% slower than desktop connections
- ◆ 4G/LTE connections... wildly depends! (-5% -- -40%)

So...

- ◆ ... Every site pays a ***time price*** when is seen on a smartphone
- ◆ Question: and so what?



The «intuitive» answer...

- ◆ We know that already, having spoken about the importance of time for users (timers etc.)
- ◆ So if a page takes 40% more, it takes more time out of any timer

But...

- ◆ ... the problem is more serious:
- ◆ It's not only a matter of session timer or global timer, this is an added delay for every single page
- ◆ And so, users compare that with the average loading times

Local time?

- ◆ In the desktop case, users wait for a maximum time ***2 seconds*** per page
- ◆ Beyond this time, they have a ***delay*** perception, and so corresponding discomfort to the site
- ◆ So, local loading time of every page is an important aspect to consider: every page should be fast enough

And in the mobile case?



- ◆ For mobile, users expectations don't change: the limit is always 2 seconds!
- ◆ And so, we have to be careful: adapting the visual layout is not enough, we also have to be careful of the possible penalites of the mobile network

So...

- ◆ ... Always take into account connection speed in the mobile cases: the best way to do it (when possible) is to have ***lighter*** pages for mobile than desktop!

Beware...



- ◆ ... that the same temporal limits also apply to *apps*
- ◆ → an app taking more than 2 seconds for an *action* is considered slow!

So for the success of an app...

- ◆ ... a fundamental factor is “***responsiveness***”: users should not perceive a delay unless the context clearly justifies that (for instance, photo upload etc).
- ◆ Note this is always valid, so also for any action that needs a network connection... (so even if it’s not our fault!)

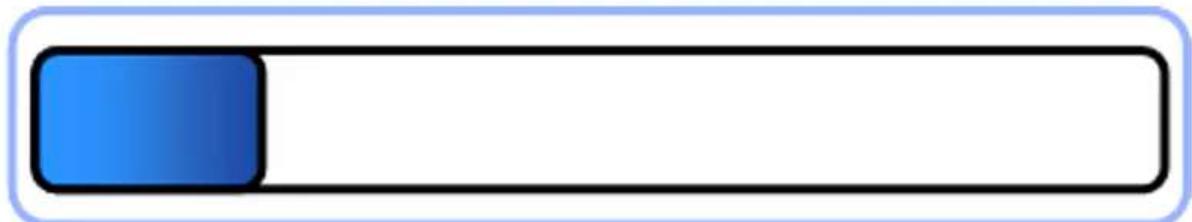
Solution 1?

- ◆ Let's learn from the desktop...
- ◆ What happens when a desktop application does something that takes too much time?

...



- ◆ We can use a *progress bar*, or a so-called *spinner*
- ◆ Intuitively: so even if there is some waiting time, users are warned



Question...

- ◆ Looks great, but what about the users?
- ◆ Answer: users don't like it!!!
- ◆ Progress bars/spinners are *explicitly* signaling a **problem**: dear user you have to wait



Metaphorically...

- ◆ Imagine being in a queue at a shop to pay, and somebody would go on and on telling us «you are on a queue, but please be patient!
- ◆ Moral: users ***perceive*** the time delay as ***longer*** when there are progress bars or spinners (!!!)

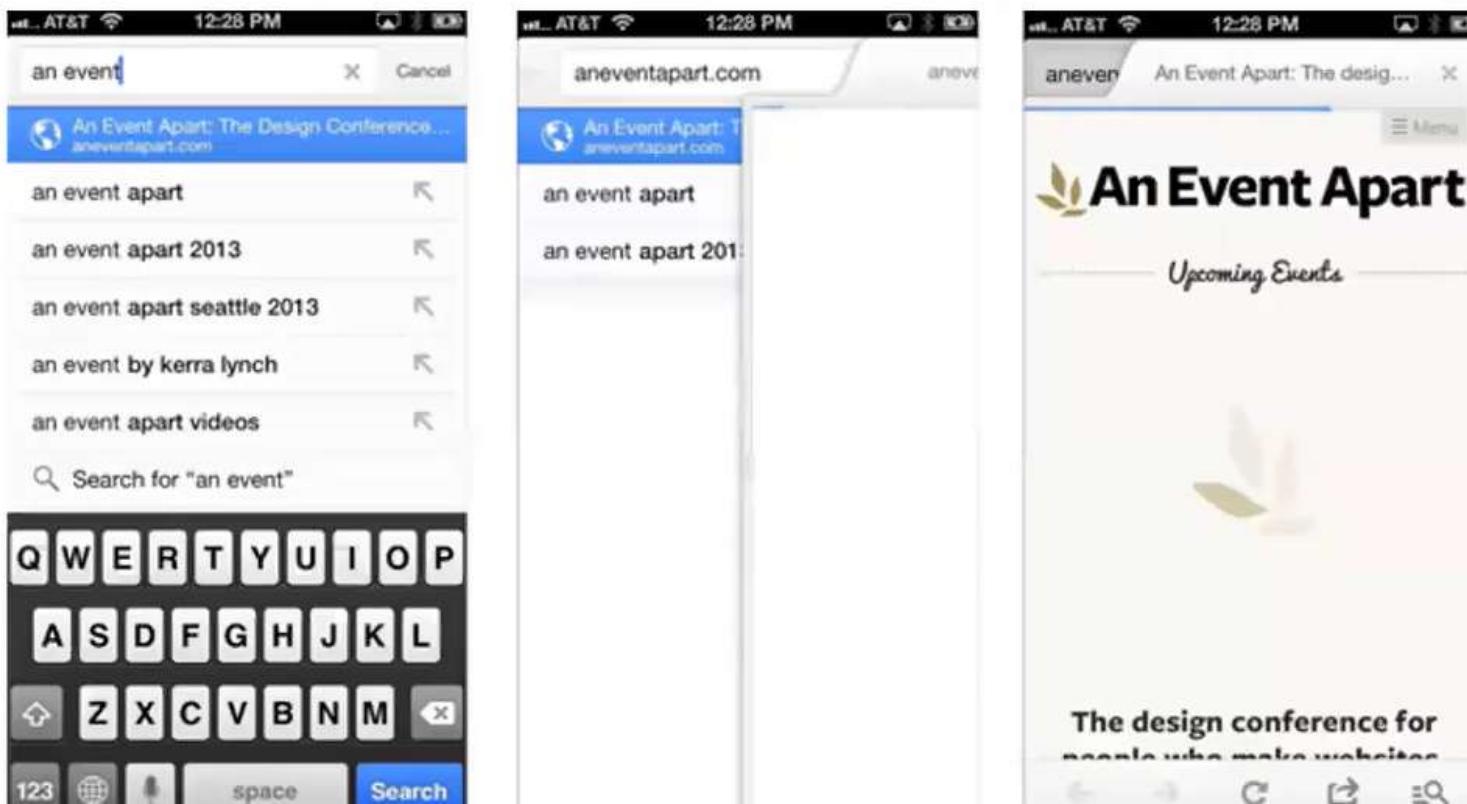


So...

- ◆ In the mobile case (site or app) these solutions are dangerous to use
- ◆ And so what?

Other techniques

- ◆ For instance, *transitioning*
- ◆ Keep the user busy, for instance with animations and so on

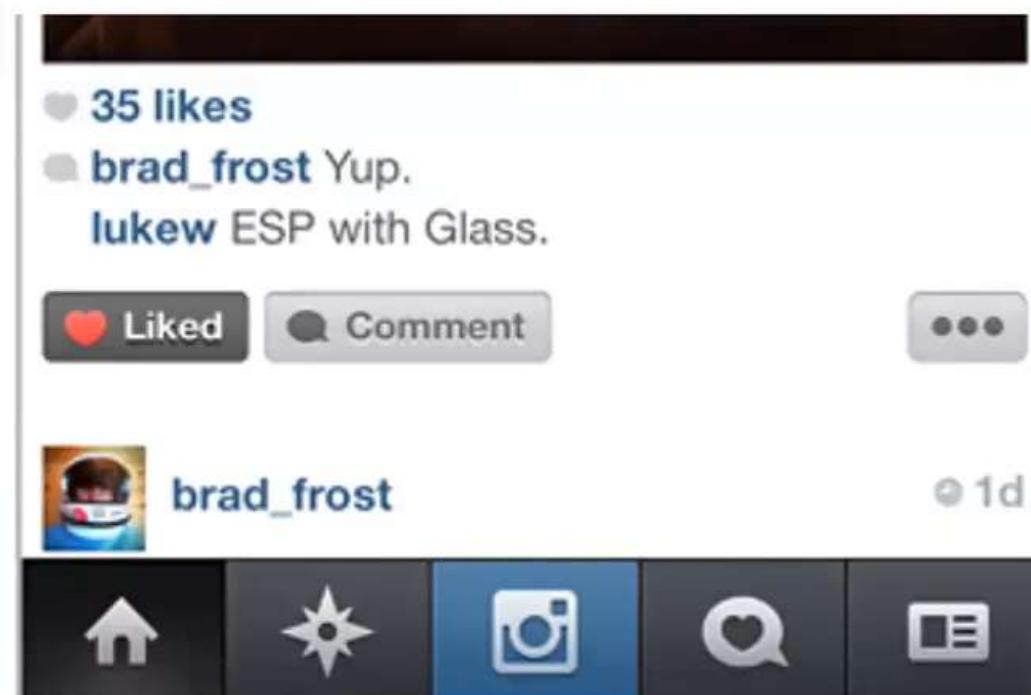
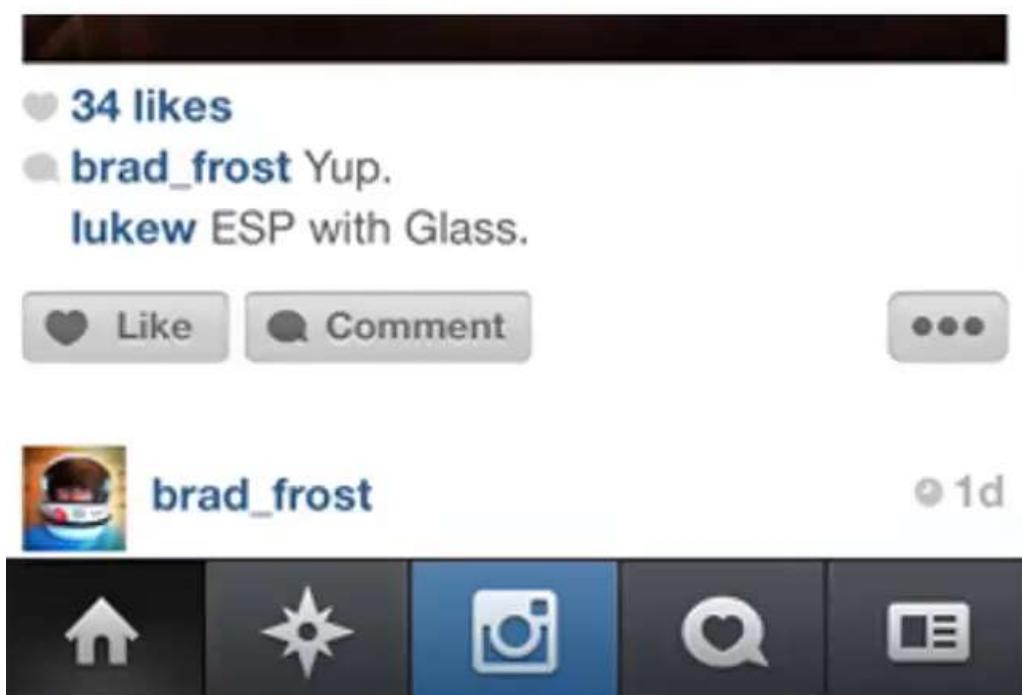


Particular case of transition

- ◆ The *skeleton screen*
- ◆ If I know the final layout of an action, I can start drawing it even if I didn't receive all the data yet
- ◆ (Remember web history, browsers evolution and progressive display: history repeats!)

Extreme example

- ◆ Instagram and the like (followed years later by Facebook...!!)



Other technique: preemptiveness

- ◆ For instance, an app has an «upload photo» action
- ◆ Classic: let the user choose the photo, then ask for a description, then upload
- ◆ Better: load the photo as soon as it's been chosen!
- ◆ → almost instantaneous upload, and users super happy!! ☺

Last remark on speed

- ◆ We left pending **0.facebook.com**: what is it?
- ◆ It's a super-fast version, limited-bandwidth, bare-bones functionalities
- ◆ So for instance, images aren't inline, and need an extra click to be seen



What??

- ◆ A super-fast Facebook,
so fast that to speed up it tries to always
use text only
- ◆ So speed is higher, but users are less
happy (more clicks etc)
- ◆ So what is the sense of all this? Why
should I offer this «0» version rather
than the normal mobile version??



\$\$\$

- ◆ When for the user the cost/benefit ratio is very low!
- ◆ The «0» version is offered free in all those places where connections are slow and quite costly



The Second Component



- ◆ The *screen size*
- ◆ Main issue: a page will hardly be seen without scroll
- ◆ So the question: how bad is scroll for mobile?
- ◆ And the answer is: it depends



In general...

- ◆ ***Horizontal*** scroll is as bad as for the desktop case
- ◆ The good news: ***vertical*** scroll instead is often not so bad

The reason?

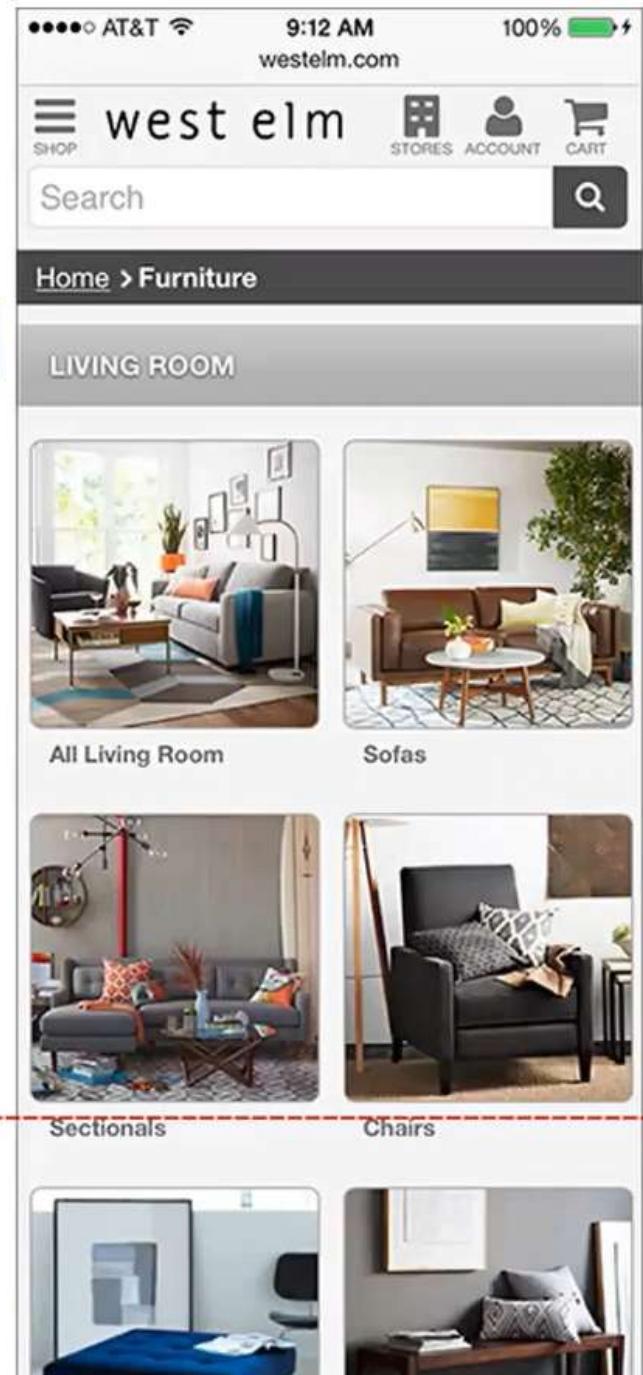
- ◆ Main reason: the physical and mental effort is quite smaller!
- ◆ Scrolling is done via ***gestures***, without the burden of using the mouse in specific zones to manage it (even worst, sometimes using dragging too)

But beware...

- ◆ Vertical scroll may also be bad for users, when it's used to offer choices (for instance menus, list of products etc)
- ◆ In such cases, the user has to keep in mind the content of what above until the choice is completed, and this generates mental fatigue (too much information in a single shot), proportional to the size of the hidden information (!)

Consequence

- ◆ It's better to minimize vertical scrolls for choice lists, and so for instance in some cases even avoid images (!)



Images in lists...

- ◆ Are justified only when it's about final products (for the same reasons already seen for the classic e-commerce case)
- ◆ This anyway can lead to the memory efforts mentioned before, and so the best solutions are anyway those that minimize scrolling

Example

••••• AT&T ⌂ 5:08 PM 44% 🔋
sportsauthority.com

SPORTS AUTHORITY. STORES CART

SEARCH

SHOP CATEGORIES + SHOP BRANDS +

Home > Athl... > Me... > Training

18 Total Items View:

Filter Your Results ▾ Sort: Top Seller: ▾

NIKE Men's Free Trainer 5.0 Cross-Training Shoes

NIKE Men's Free Trainer 7.0 Super Bowl Cross-Training Shoes

••••• AT&T ⌂ 5:08 PM 44% 🔋
sportsauthority.com

SPORTS AUTHORITY. STORES CART

SEARCH

SHOP CATEGORIES + SHOP BRANDS +

Home > Athl... > Me... > Training

18 Total Items View:

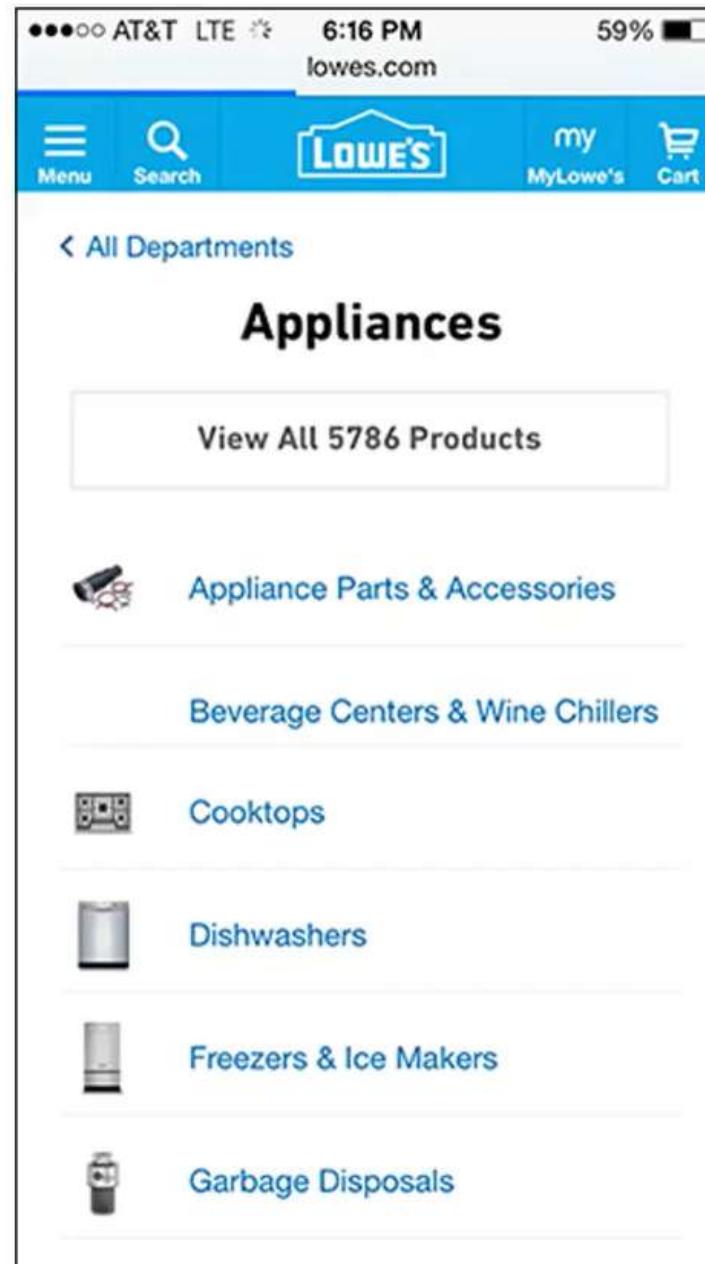
Filter Your Results ▾ Sort: Top Seller: ▾

NIKE Men's Free Trainer 5.0 Cross-Training Shoes
\$95.00
More Colors Available
★★★★★
Free Returns on Shoes!**

▲ Top

And the middle way?

- ◆ For instance, text with small images?
- ◆ That's good for general categories/menus, but not for final products: the primary pulsion is always to also ***see the product***



Other consequence of the smaller mobile size

- ◆ To remedy for the small size, one solution can be to use *icons* rather than *text* for buttons
- ◆ The problem is the same as for the desktop though: this works if users know that the icon means (so ok for instance to the search lens, but others?)

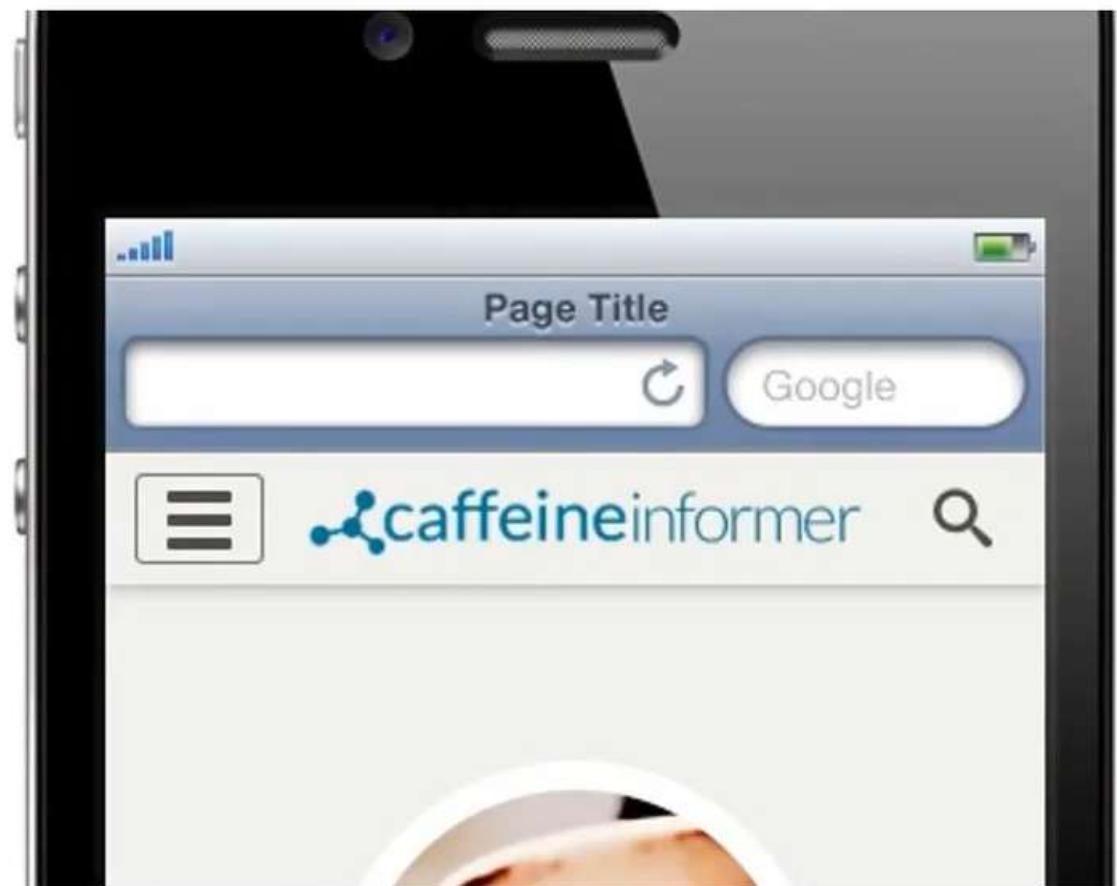
Consequence

- ◆ Unless the icon is super well-known, users always prefer ***text*** to icons!

Example...



The Hamburger!



The big problems of menus for mobile

- ◆ Menus are an essential components of sites, but that's potentially so wasteful for mobile: we have to use lot of text in a small screen
- ◆ BUT... if we are strong enough we can «push» in some other way...(!)



Example

- ◆ Firefox changed some time ago its desktop interface, introducing the ***hamburger***, on its ***right*** side (!)
- ◆ → ***users*** got ***super-angry*** (!!)
- ◆ So why??



Answer

- ◆ Firefox -> Mozilla Corporation
- ◆ Corporation = “for profit” private entity
- ◆ Financed by whom?
- ◆ Answer: since many years, primarily by Google (with hundreds of million dollars...!)
- ◆ This is way for instance Firefox has Google as default search engine (!!)

Moral

- ◆ Given the huge benefits of the hamburger for mobile sites and apps (Android world), Google «accelerated», changing its and others desktop interfaces from desktop to mobile...





Back to icons

- ◆ If we want to use icons the best is to also use text (like for desktop), but if we really want to use an icon without text...
- ◆ We should at least respect the **explainability** principle:
- ◆ Keeping pressed the icon the user can obtain textual information on its action



Moreover...

- ◆ A companion principle should also be always used: **escapability**
- ◆ --> a user can always «escape» from an action just by moving away the finger from the icon
- ◆ Principle valid in general for any touch action!
- ◆ (There are then also other factors related to icons and clickable objects due to the Third Component that we will see soon)

Other consequence of mobile size...

- ◆ Having a smaller screen implies also a redefinition of what «invasive» means for an advertisement banner

Advertisement sizes

- ◆ We already mentioned it in the classic case, saying they should be «too big»
- ◆ This is intuitive enough for a desktop screen size, but what about mobile screens?
- ◆ Let's see the most common sizes

Interstitial ads

- ◆ Fullscreen,
HTML5-enabled



Smart banners

- ◆ Sizes:
(screen width) x (32/50/90)



Users...?

- ◆ Interstitial ad: generally bad, but something is even worst
- ◆ Smart banners that are implemented in a ***fixed*** (not scrollable) position
- ◆ An ***always-visible ad*** is a good idea, but that annoys users! It takes away precious screen room, and if it's ***changing*** even ***worst*** (remember the blink effect etc!!)



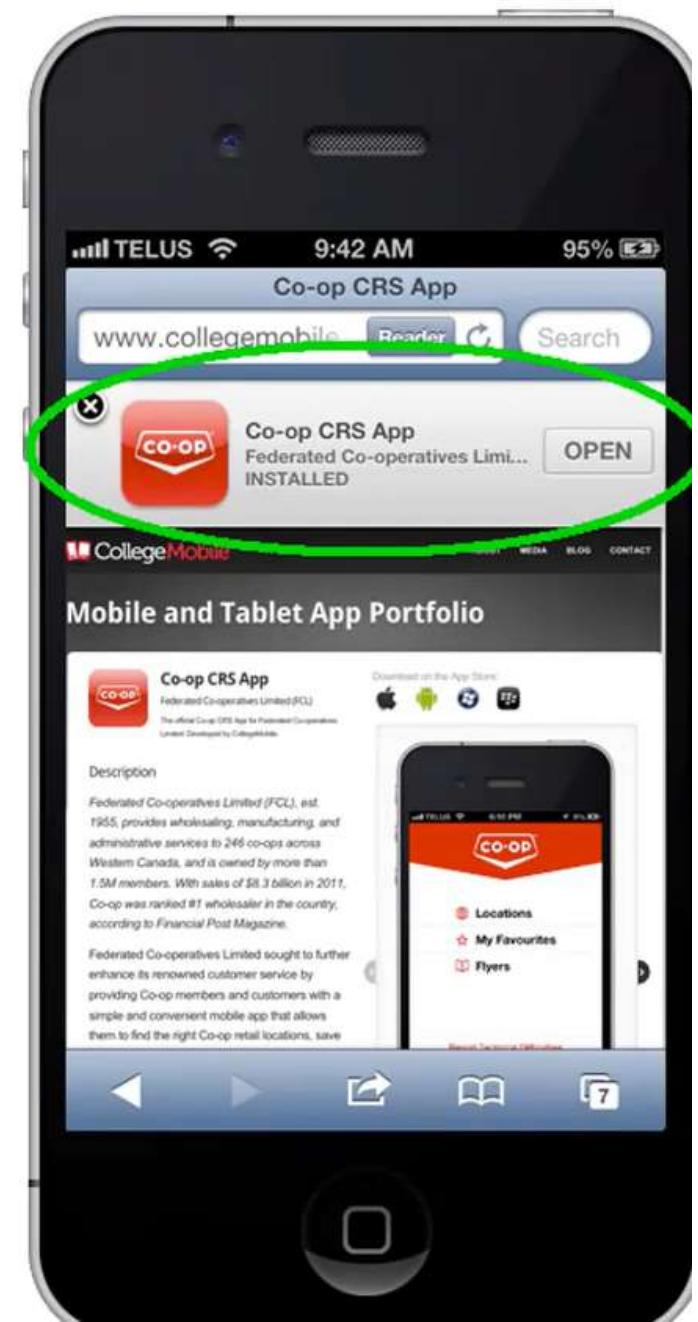
Moreover

- ◆ Small banner → temptation to grab people attention: that's the «Dancing Jesus»!!!
- ◆ Obviously that's **TOTALLY ANNOYING** for users (desktop rules apply here too!)



Last but not least...

- ◆ The temptation of the so-called Smart App Banner (publicize the site app)
- ◆ Result: *super annoying* for users (equivalent to the pop-up for the desktop...!)





III

A large, stylized number 'III' with a cracked, stone-like texture, located in the top right corner of the image.



The Third Component

- ◆ The interaction way: *fingers*
- ◆ Fingers have pros and cons
- ◆ Let's start from the positive aspects



Drag



- ◆ Unlike the desktop case, *drag* is not problematic: there is no muscle effort to keep pressed a mouse...
- ◆ This aspect make gestures particularly appealing to users (remember gestures could after all be done also in desktop with a mouse...!), and related uses (like swype)
- ◆ Just be careful to respect an essential principle...

Untiming



- ◆ Actions activated by finger pressing should be dependent on duration, but just distinguish between «tap» and «drag» (prolonged press)
- ◆ Some mobile user interfaces/apps instead assign different actions to different press durations: this causes ***errors*** and high ***user stress***

Let's pass now...

- ◆ To the problem with fingers
- ◆ Major problem: compared to the mouse, fingers are rather rough
- ◆ They are usually referred in the mobile world as "***fat fingers***" (!)

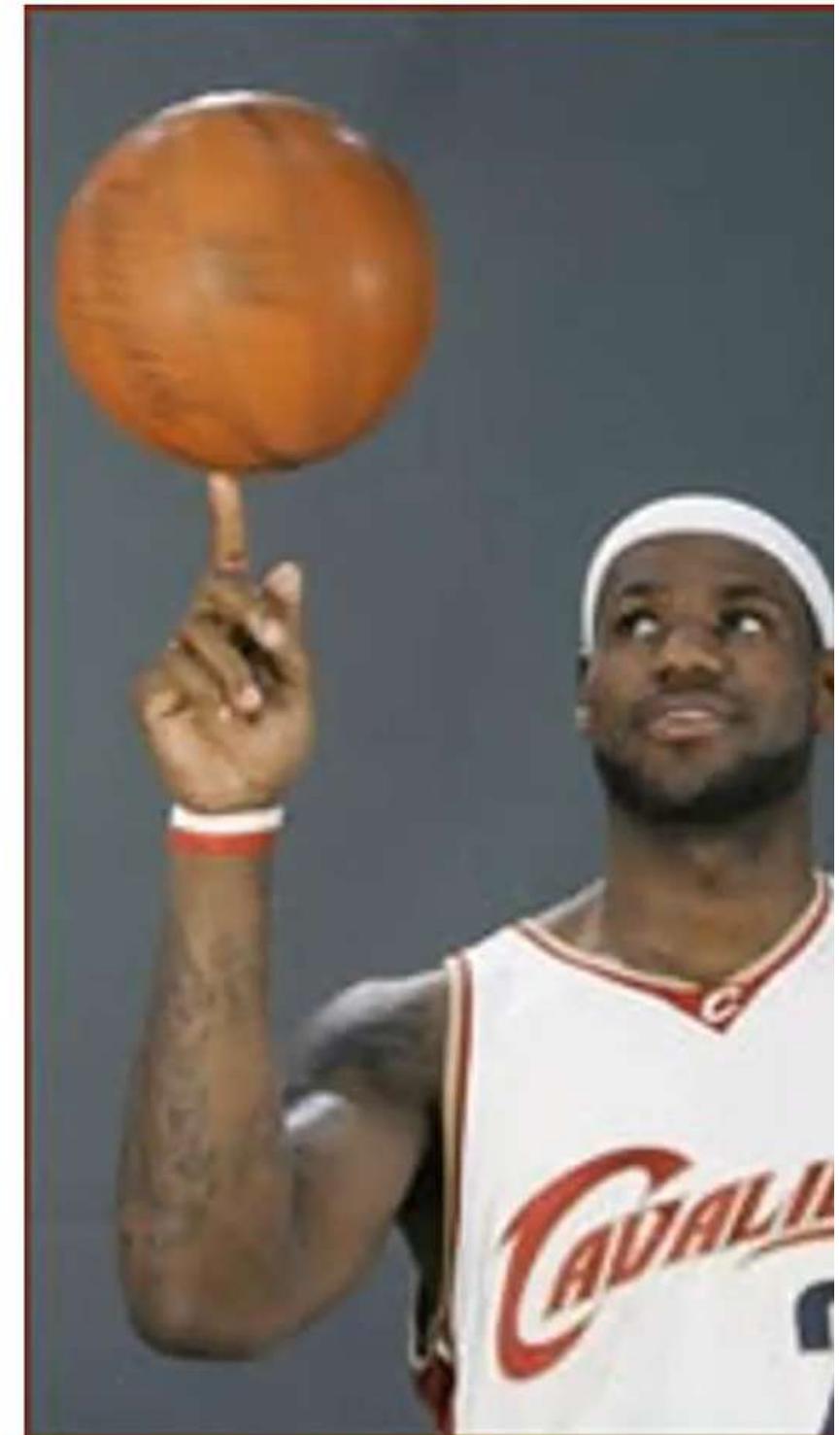
What happens in reality...

- ◆ Is that often fingers are the cause of clicking problems
- ◆ To understand how to solve them, we need to better understand our fingers...



“Fat”...?

- ◆ To know whether our buttons are too small, we should know how big fingers generally are...



The answer



- ◆ The average finger is 11 millimeters wide (!)
- ◆ Children fingers? 8 millimeters (!!)
- ◆ Big fingers? Up to 19 millimeters (!!!)
- ◆ Moral: any clickable area must be big enough so to be well centered with a finger!

Consequences

- ◆ The ***minimum size*** of any clickable is **7x7 millimiters**
- ◆ Around, we need a «padding» safety zone wide at least **2 millimiters**
- ◆ If we really don't have room, we could go down to **5x5**, factoring a 20% decrease in precision (→ user frustration)
- ◆ If on the other hand we want to make users happier, we can use **9x9** millimiters and beyond



In practice...

- ◆ It's amazing the number of mobile web sites and of apps that don't respect these basic guidelines, and so has severe usability problems (!!)
- ◆ To partly diminish these problems, we can use a principle which is anyway important in itself

Reversibility

- ◆ Every action taken should be reversable (even more in situations of potential errors due to the fat fingers)
- ◆ Useful principle also in connection to ***explainability*** (so users can always safely try to see what a button is doing, having no permanent damage)

Example of epic fail...

