

COM3504/6504 The Intelligent Web

Prof. Fabio Ciravegna
Professor of Pervasive Computing
Department of Computer Science
The University of Sheffield
f.ciravegna@shef.ac.uk

© Fabio Ciravegna, University of Sheffield

The Module (ctd)

- Aspect 2: General trends that help understand the evolution of the Web across the years
 - so as to be able to guess potential future developments
 - how the Web is at the core of a general development in the Digital World
 - which is (becoming?) pervasive in our lives.
 - how the Web started from an application running on an office machine (a computer)
 - to turning into the backbone for mobile devices, self-driving cars and the Internet of Things.
 - exploring the socio-economic aspects of Web big data in search engines and the social Web,
 - discussing how our lives are being revolutionised by the always-on Digital World supported by the Web
 - discussing e.g. why Google took the world by storm, where they get their money from and why the long tail makes them \$12B a year
 - Discussing the disintermediation of the Web by Tech giants and the impacts on our lives
 - Discussing the political impact of Web Technologies
 - including threats to our democracy, Brexit and Trumpism

© Fabio Ciravegna, University of Sheffield

3

The module

- Explores advanced methods for Web Technologies. It covers two aspects.
 - Aspect 1: Advanced aspects of programming the Web:
 - from asynchronous and bidirectional client-server architectures, to the creation of Progressive Web Apps for mobile phones, to methods for teleconferencing using a browser, to recommender systems.

© Fabio Ciravegna, University of Sheffield

2

Prerequisites

- This is an advanced module on Web technologies
 - A solid knowledge and facility with basic Web technologies like HTML, CSS and Javascript are a requirement.
 - You are expected to be hitting the ground running on these.
- If you do not have this knowledge you should consider if you are in the right module
 - For MSc students, a set of slides are provided to catch up with the above

© Fabio Ciravegna, University of Sheffield

4



- The module develops over 10 weeks. Weeks 1-8 will be before Easter and 9-10 will be after the break
- Each week we will have 2 hours of lectures on Thursdays and two hours of lab classes on Fridays
- Each week we will cover one specific topic.
- The two lab classes will be sometimes completely dedicated to our lab exercises but sometimes the first hour may be dedicated to parts of the lectures
- Also it is possible that some weeks you will feel that the second hour of lab classes is not necessary.
 - Feel free to leave the virtual room in case.
- In my experience, each cohort of students and even each student start the module with a different knowledge level of Web technologies.
- For some of you some topics will be already largely known while other topics may be completely new.
- I will have to teach all parts as they were completely new



The three parts of the programme

• Part 1: Horizon scanning

- **Lecture 1:** introduction to the Digital World. Why the web is not what you thought it was
- **Lecture 10:** the future: how digital tech are influencing our lives and how you can learn to navigate the future. Be bold!

• Part 2: advanced client-server communication

- **Lecture 1 and 2:** introduction to the basics of Node.js, asynchronicity in Javascript and data exchange using JSON
- **Lecture 3:** beyond stateless client-initiated synchronous communication
 - bidirectional asynchronous communication (Ajax, socket.io)
- **Lecture 4:** client side persistent in IndexedDB, caching via service workers, Progressive Web Apps
- **Lecture 5:** server side persistence at Web scale: noSQL databases



The three...

- Part 3: making the web intelligent
 - **Lecture 4:** Collective Intelligence
 - **Lecture 6 and 7:** Collective intelligence: the Social Web, social media recommender systems, teleconferencing via browser
 - **Lecture 8:** Collective intelligence over very large scale (search engines)
 - **Lecture 9:** The Web of Data (knowledge graphs, Wikidata, etc.)



Week 1 Introduction

- Week 1:
 - The Digital World: how did we get here and where is our society going? The role of Web technologies in the big picture beyond an application running in the browser. How to create and run pervasive Web technologies.
 - Introduction to client-server architecture in Node.js
 - Lab class:
 - getting acquainted with the development environment (IntelliJ)
 - Creating your first client-server architecture using Express



Week 2: Asynchronicity in Javascript and NodeJS

Availability: Item is available, but some students or groups may not have access.
Enabled: Adaptive Release

- We will cover the way to implement asynchronicity in Javascript and NodeJS
- We will explore the issue of complexity of callbacks (callbacks hell) and how to reduce it
- We will see Javascript's Promises as a way to escape the callbacks hell
- We will see Server to Server communication in nodeJS
- We will briefly mention the issue of CORS which prevents posting to a server that is not the one that has served the HTML file



The
University
Of
Sheffield.

Week 3: Asynchronous Bidirectional Client-side Interaction

We will cover a client server interaction methodology that overcomes some of the limitations of the standard http protocol. In particular we will see:

- asynchronous data exchange between a browser and a server using Ajax
- asynchronous bidirectional client/server communication using Websockets (socket.io)
- browser based peer to peer video communication using WebRTC

At the end of this lecture you should know:

- Why we need an event driven, flexible client server interaction with callbacks also in the communication between browser and server
- How to use Ajax (via JQuery)
- How to create a callback for Ajax to act on the results
- how to create bidirectional client/server architectures using [socket.io](#)
- how to create a room in socket.io and have different clients in different rooms
- how to build a simple chat system

© Fabio Cravagna, University of Sheffield

9



The
University
Of
Sheffield.

Week 4: IndexedDB and Service Workers

In this lecture we will see:

- how to store large amount of data in a database running on the browser (IndexedDB)
- how to provide a full offline experience using Service Workers

These are two functionalities that make the browser very powerful. While in the past the browser was a simple interpreter of HTML/CSS/Javascript, now it becomes an application that can run offline and persist large amount of data.

Nest week we will see how to use these to create a Progressive Web App which is a website that can be turned in something very similar to a native mobile app, because it works offline and provides notifications even when not in use.

Week 5 PWAs and NoSQL Databases

In this lecture you will learn about Progressive Web Apps

The we will introduce noSQL databases and in particular we will focus on MongoDB

Week 6: Collective Intelligence

In this lecture we will:

- learn how to document a nodeJS API using openAPI
- introduce the concept of collective intelligence and analyse how it applies to the social web
- learn to create a collaborative system

The lab class will concern the use of Swagger to document your APIs

© Fabio Cravagna, University of Sheffield

10



The
University
Of
Sheffield.

Week 7: Collective Intelligence from Social Media

In this lecture I will show how the technologies we have studied in the module so far allow to create complex system analysing social media for emergency response.

The system is a real world system used in the emergency control rooms for events involving over a million people

There is no specific exercise for the lab class. The lab class will be dedicated to support for the assignment



The
University
Of
Sheffield.

Week 9 The Knowledge Graph

In this lecture I will introduce the knowledge graph and show how data can be connected through the web, so that we can have data that is linked across websites and databases. This makes the data and knowledge browsable exactly like we can browse web pages. We will see how to use it in industry and how it is used by the search engines.



Week 10 - The Future

We will try to see what the future holds for the web and how you can make a career in working on the web

However Lecture 9 may be replaced by 1:1 feedback on the 1st part of the assignment, so we may compress some of the topics in the previous lectures to make sure to cover the knowledge graph

© Fabio Cravagna, University of Sheffield

11

© Fabio Cravagna, University of Sheffield

12



Reading Material

- Lecture notes
- Lots of Web documents to read
 - This is module on Intelligent Web!
- Any book about HTML, Javascript, MySQL and PHP will do
- Join the Facebook group:
 - <https://www.facebook.com/groups/652598341469951/>
 - Where you and I can post interesting articles, news etc.
 - Not an official module resource, though (use MOLE!).
 - It is also to keep in touch after the end of the module
 - in case there are relevant news - e.g. job opportunities I become aware of

© Fabio Cravagna, University of Sheffield

13



Assessment

- One project will constitute:
 - COM3504: 100% of the assessment
 - COM6504: 80% of the assessment (20% is for a Mole quiz - see below)
 - Released at end of Week 4
- Two submissions:
 - First part of assignment (formative feedback - marked - submission not required)
 - Deadline: Friday of week 7 or 8 (to be confirmed)
 - Submission via Mole
 - 1:1 feedback in Week 9 (no lab class on that week)
 - Informal marks
 - Final submission
 - The whole assignment (including the revised part 1)
 - Deadline: Friday of week 12 at 23:59:59 (to be confirmed)
- For COM6504:
 - Mole Quiz on basic web technologies (HTML, Javascript, CSS) on Friday of week 4

© Fabio Cravagna, University of Sheffield

14



Groups

- Assignment is to be done in groups
 - Groups must be composed of a **3 people**
 - No groups of <3 or >3 are allowed **for any reason**
- You can suggest your group
 - However I am required to choose. This means that
 - I may change the group composition or pair people together
 - if they are unable to find a suitable group
 - I will add a third member to any group of two

© Fabio Cravagna, University of Sheffield

15



Groups, oh no!

- Every year at least 3 groups have serious problems during the first assignment
 - I have heard this several times:
 - Person 1:
 - My partner
 - did not contribute at all
 - Was never present when we decided to meet
 - Person 2:
 - I was busy, my cat was ill, I had a dentist appointment, I actually attended once
 - Can you tell me where the problem is?

© Fabio Cravagna, University of Sheffield

16



Groups? Oh no!

- Groups of 3:
 - 2 member says:
 - Our partner did not contribute at all
 - He is always silent and sometimes he does not even come to the meetings
 - The other member says:
 - My partners work very fast, talk a lot among themselves and never let me understand what they do. They are probably better than I am
- Can you tell me where the problem is?

© Fabio Ciravagna, University of Sheffield

17



Groups? Oh No!

- One member says:
 - I have worked for many years as software developer in companies, I know how to develop a project. I will set out a plan for everybody by using a well established strategy that I have devised myself based on best practices
- The other members say nothing but in a while stop coming to the meetings. They are simply lazy and not committed to their study
- The group splits (yes you can split a group)
- Who do you think got the best marks?

© Fabio Ciravagna, University of Sheffield

18



Groups? Yes, groups

- Select your partner(s) with care
 - Ask them about their marks
 - Do it!!!
 - Choose someone with marks similar to yours
 - Do not try to be with better people to try to scrape better marks
 - it does not work!
 - Do not chose someone just because you fancy them
 - Find a boy-/girl-friend in another way
- Be very careful when you work in a group
 - Listen to any sign of distress or disinterest
 - Talk to your partner frankly and honestly
 - Discuss with me any doubt
 - That is not being nasty to your partner!!

© Fabio Ciravagna, University of Sheffield

19



What if I cannot find a group?

- Start looking for a group NOW!!!
- Ask around
- Socialise
- Fill the appropriate form
- Talk to me asap
 - We will have a dating session for people who are without group
- IF ALL FAILS you may be required to do the assignment on your own
 - Please make sure that this does not happen

© Fabio Ciravagna, University of Sheffield

20



Start forming your group proposal
TODAY!!

Every year at least one person is left out, ends up doing the assignment by themselves (and generally fails)

Every year at least two groups are totally dysfunctional because they did not choose with care/were late/were just checking if partners were good looking

Elastic Circumstances | Institute of Sheffield



Register your group NOW!

- Every year 3 or 4 people cannot find a reasonable group because they left this to too late
 - One person always ends up doing it on their own
 - THIRD YEARS cannot group with MSc students

<https://docs.google.com/spreadsheets/ccc?key=0ApRYU-DxjhLDdGVMb1IDd083MVJnSC13OGkwdkZDd2c&usp=sharing>

A52		A	B	C	D	E
1	Please note that MSc students cannot be in a group with undergrad students	Are you a single person looking for a group? Please use tab no. 2				
2	Group Name	Member 1	Member 2	Member3	contact emails for the group	
3						
4						

© Fabio Ciravegna, University of Sheffield

22



Looking for partners?

- No worries. Register now (second sheet)
 - And contact the available people immediately!!

Fabio Ciravegna | University of Sheffield

23



How to pass

- Come and listen to the lectures
 - If you do not understand, tell me
 - Feel free to ask questions at any time on Thursdays
 - Come to the lab classes and do all the exercises immediately
 - Most lab classes give you a piece **to use in the assignment**
 - Complete the assignment every week

Fabio Ciravagna, University of Sheffield

24



The
University
Of
Sheffield.

Some past experience

I get 5 big assignments at the same time now in the second semester. It's really too difficult to handle all assignments well for me because the time seems too limited

I feel that two hours of lectures were not enough to cover the subject in the detail I would of liked.
Please increase it to 3 - 4

(this year we have increased the lab classes)

© Fabio Cravegna, University of Sheffield

25



The
University
Of
Sheffield.

Assignment too intensive

- Arguably the best module I have taken throughout my time at uni.
- That said,
 - The workload for the assignment was grossly underestimated.
 - The learning curve for NodeJS is very steep, and the "30hours" work per person estimate is miles off.
 - I'm in a team of three people, all of us have comfortably performed to a first class standard throughout our entire degree
 - and we easily spent around 60+ hours each on the initial stage of the assignment.

The goals of the assessment is clear but overkill and requires as much work as a dissertation coding

© Fabio Cravegna, University of Sheffield

© Fabio Cravegna, University of Sheffield

26



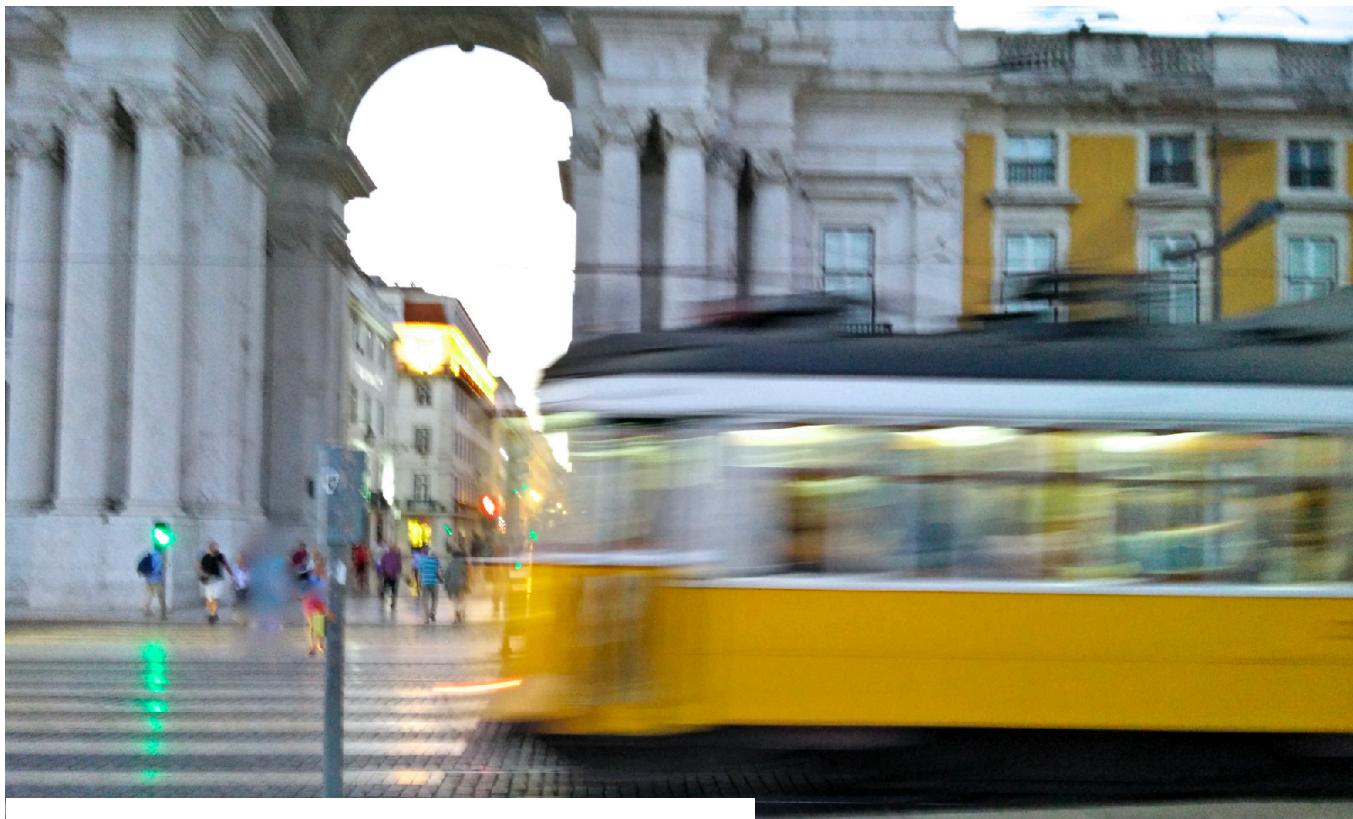
Well, don't do that!

- The assignment is largely open ended.
 - Please keep it in proportion to the credit value
 - Do not try to build a system with infinite features
 - Just make an excellent work by keeping to the letter of the assignment
- (That said, some of last year's solutions were fantastic!)

© Fabio Cravegna, University of Sheffield

Questions?

© Fabio Cravegna, University of Sheffield



COM3504-COM6504 The Intelligent Web

Professor Fabio Ciravegna
Professor of Pervasive Computing
Department of Computer Science
University of Sheffield
f.ciravegna@shef.ac.uk



Then by applying linearly these laws

	2022	2027	2032	2037
Networks	500M	3G	16G	96G
Computers	£600	£3,600	£19,200	£115,200*
Disks	1T	6T	32T	192T

* The equivalent to 192 computers

You do not believe it?

© Fabio Ciravegna, University of Sheffield

3

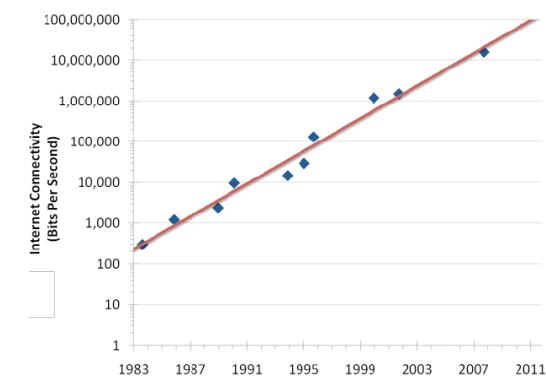
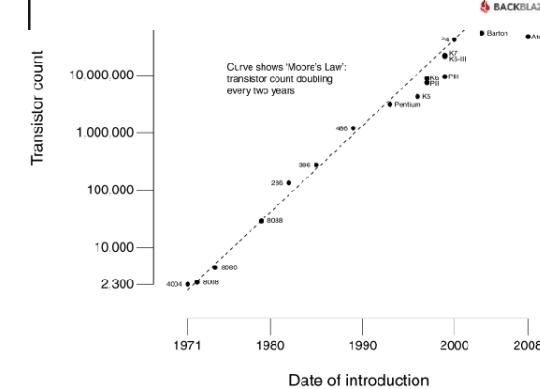
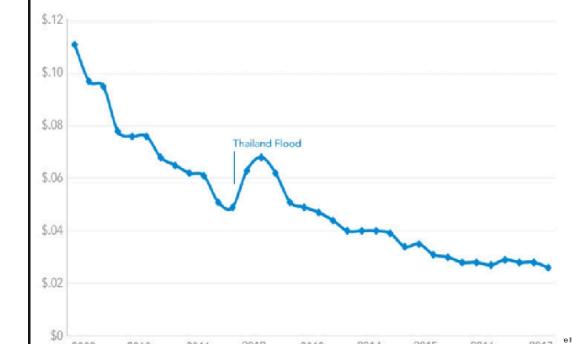


Moore's, Nielsen's and Disk Space Law

Computer power, speed of broadband and disk storage capability doubles every year (same cost)

The same computer power, speed of broadband and disk storage capability will cost half the current price in 2 years' time (and basically will use half space)

Backblaze Average Cost per GB for Hard Drives
By Quarter: Q1 2009 - Q2 2017



© Fabio Ciravegna, University of Sheffield

2



50 years ago and today



Source: <https://www.slideshare.net/sqrajper/mobile-computing-24722802>

© Vita Lanfranchi, University of Sheffield

4

Technology Cycles – Still Early Cycle on Smartphones + Tablets,
Now Wearables Coming on Strong, Faster than Typical 10-Year Cycle

Technology Cycles Have Tended to Last Ten Years

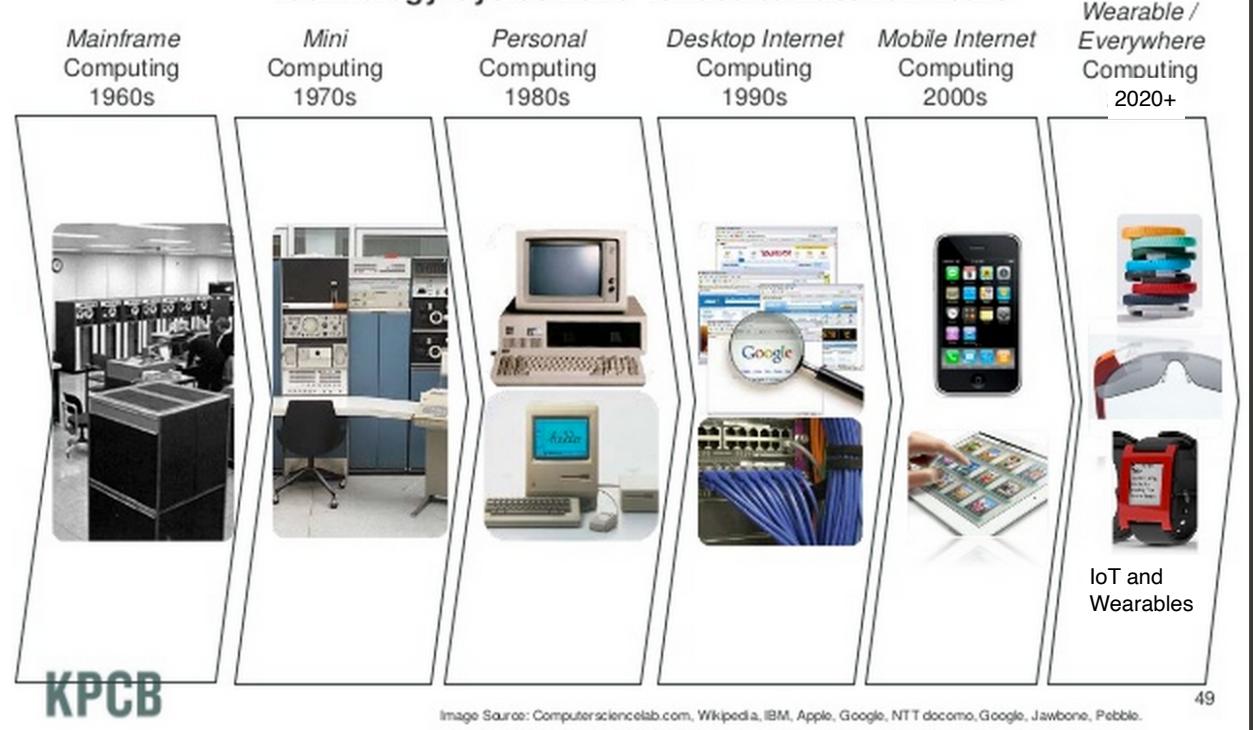


Image Source: ComputerSciencelab.com, Wikipedia, IBM, Apple, Google, NTT docomo, Google, Jawbone, Pebble.

© Fabio Ciravagna, University of Sheffield

49

49 /117

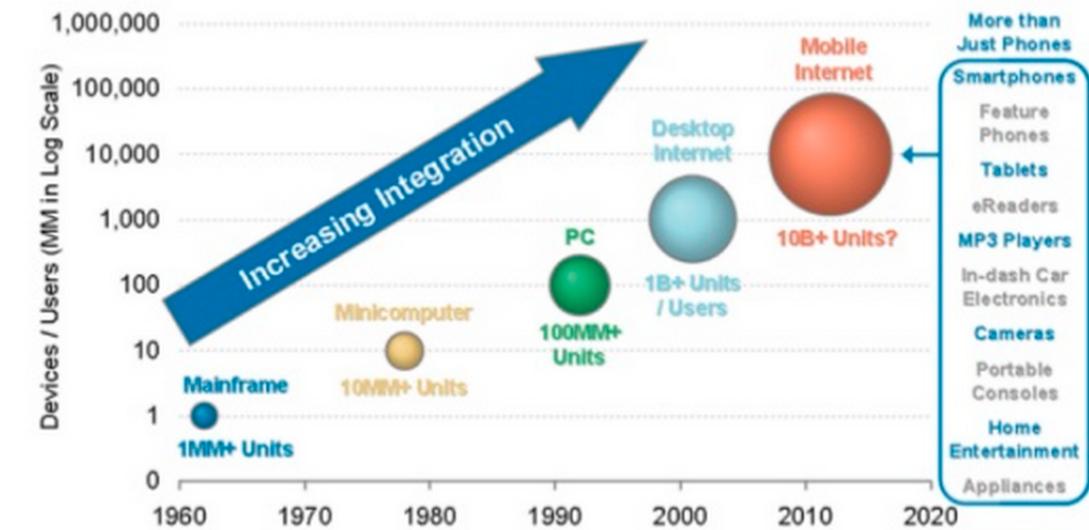
5

**Each New Computing Cycle =
10x > Installed Base than Previous Cycle**

Exhibit 29

Each new computing cycle typically generates around 10x the installed base of the previous cycle

Devices or users in millions; logarithmic scale



@KPCB

Source: Morgan Stanley Mobile Internet Report (12/09)

<http://www.kpcb.com/internet-trends>

11

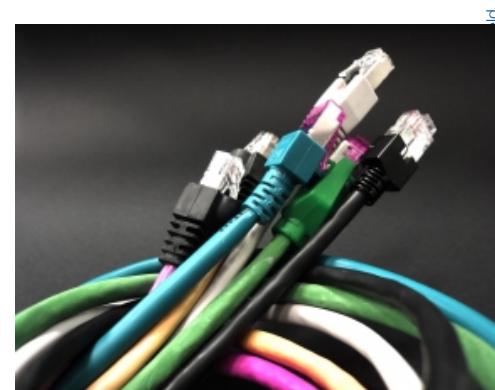


The Digital World

Building a Digital Knowledge Economy in the 21st Century will be fundamental to the UK's future prosperity. For the country to reap the maximum benefits, we need to put people at the centre of all our digital thinking.

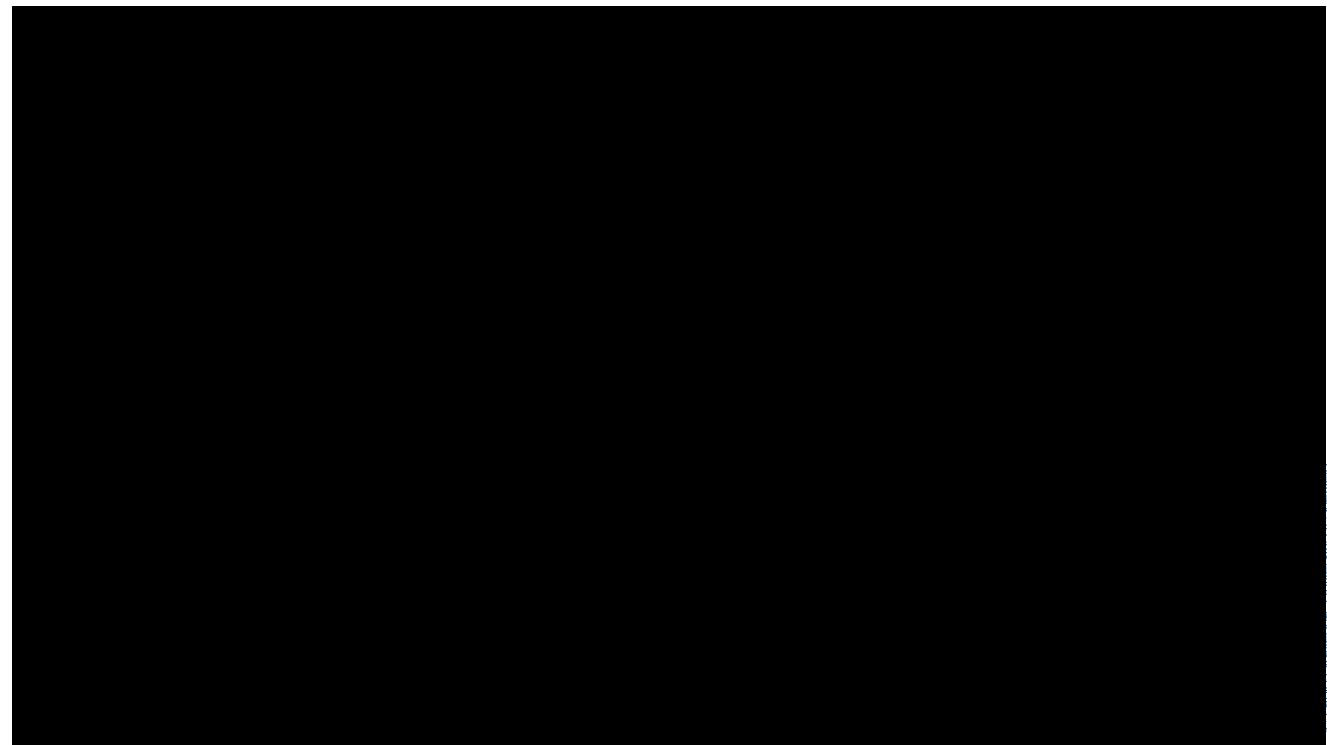
Digital Britain, Final Report, June 2009

- Analysis of requirements of real users
- Development and use of Digital Technologies
- Study of their impact on the society
- (repeat)



No longer office machines

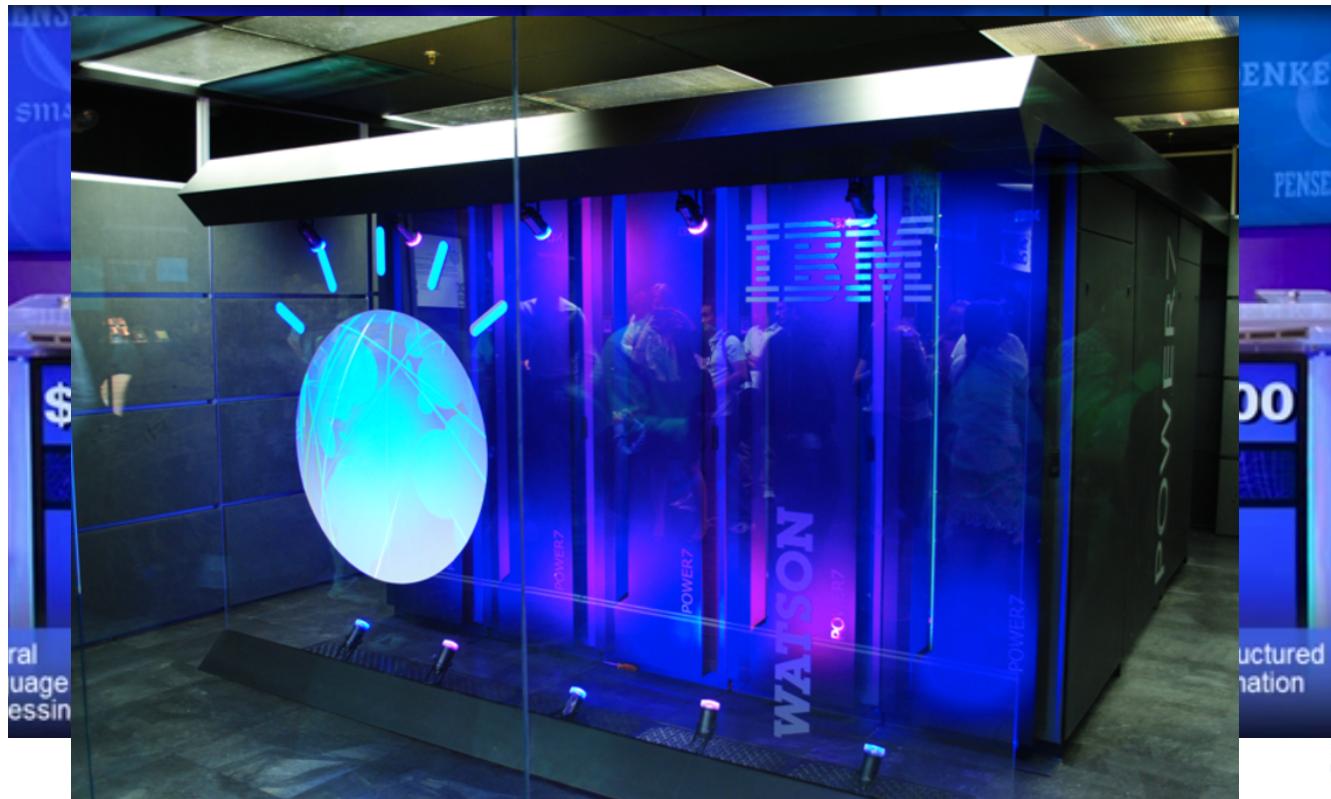
in my eyes (not an office machine)



What is important here is the idea, not the actual product (which failed from a commercial point of view btw)

9

Watson



11

amazon echo
amazon.com/echo



10

You as focus

- The digital world is not (only) about developing digital technologies
- It is about **PEOPLE** having those technologies
 - With you
 - Around you
 - 24/7
- It is about those instruments communicating
 - To create a global communication system
 - To provide the **right information** at the right time



The
University
Of
Sheffield.

The shape of things to come

And yes before you ask, these are all web technologies
(surely you were not thinking of studying Web pages in this module?)

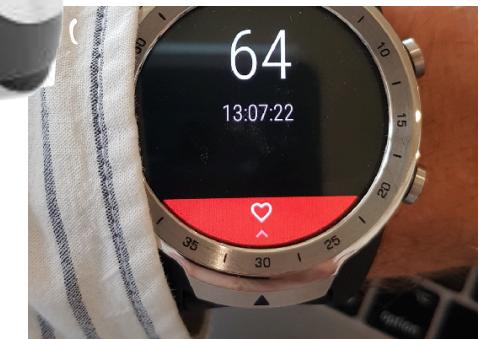


© Fabio Ciravagna, University of Sheffield

https://en.wikipedia.org/wiki/The_Shape_of_Things_to_Come



It is about Health



Self driving cars



It is about the Economy

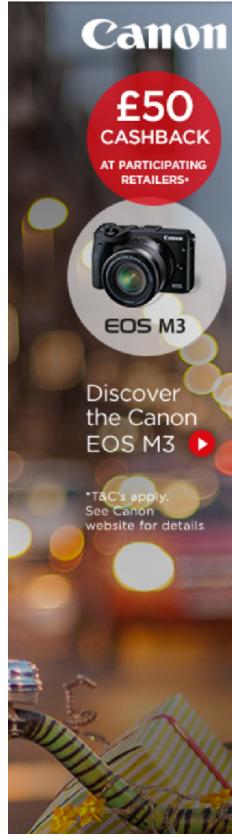


© Fabio Ciravagna, University of Sheffield

14

© Fabio Ciravagna, University of Sheffield

16



The death of the cabbie? Uber wants to buy 500,000 self-driving cars

CEO wants to hoover up all of its stock

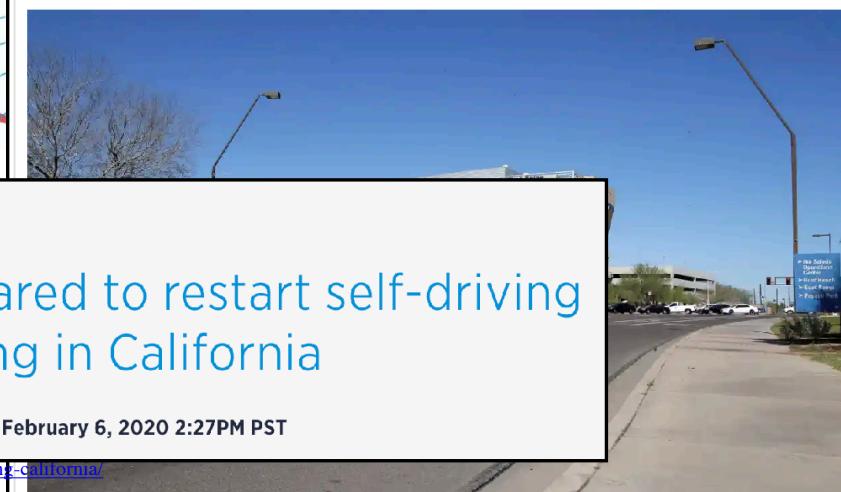
Christopher Hooton | @christophhooton | Wednesday 8 July 2015 | 0 comments



sl

Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian

Tempe police said car was in autonomous mode at the time of the crash and that the vehicle hit a woman who later died at a hospital



News

Uber cleared to restart self-driving car testing in California

By Allison Matyus February 6, 2020 2:27PM PST

<https://www.theguardian.com/>

<https://www.digitaltrends.com/news/uber-self-driving-testing-california/>



REVIEWS NEWS VIDEO HOW TO SMART HOME CARS GAMES DOWNLOAD

CNET > Sci-Tech > How self-driving cars will cut accidents 90 percent (Q&A)

How self-driving cars will cut accidents 90 percent (Q&A)

For Road Trip 2015, CNET talks with the University of Michigan's Peter Sweatman about the rapid merging of computers and cars, and the fake city in Ann Arbor where it's being put to the test.



Sci-Tech



by Stephen Shankland

ANN ARBOR, Michigan -- Peter Sweatman isn't in charge of the computing revolution that's sweeping the auto industry, but he's at the center of it.

As director of the University of Michigan's Transportation Research Institute (UMTRI) in



19



Nah! It will never happen

Robert Thurston, a U.S. steam engine expert, opined in 1894, no less, that horses are not only "self-feeding, self-controlling, self-maintaining and self-reproducing, but they are far more economical in the energy they are able to develop from a given weight of fuel material, than any other existing form of motor."

Car propaganda also portrayed the horse as "untamable beast" and author of "frightful accidents." At the same time motor enthusiasts railed against regulations, speed limits and licensing requirement for new fangled jalopies.

In the end the removal of the horse from urban life and later the farm became a protracted drama that took more than 50 years. It also required the messy adoption of three fossil-fuel technologies.

<https://thetyee.ca/News/2013/03/06/Horse-Dung-Big-Shift/>

© Fabio Ciravagna, University of Sheffield

18



It is about Education



GlobeNewswire

Global Kids Tablet Market Will Reach USD 36.89 Billion By 2027: Zion Market Research

According to the report, the global kids tablet market was USD 9.54 billion in 2018 and is expected to generate USD 36.89 billion by 2027, at a CAGR of 16.2% between 2019 and 2027.

f t in G+ @ Email Print Friendly Share

July 04, 2019 09:01 ET | Source: Zion Market Research



© Fabio Ciravagna, University of Sheffield

20



The
University
Of
Sheffield.

Wii are getting fitter: Retirement home installs computer game to keep residents trim

Last updated at 17:38 13 September 2007

Young-at-heart pensioners bored of bowls and bridge have started videogame competitions against each other at their retirement home.

Senior citizens at the Sunrise Home in Birmingham have ditched their zimmerframes because they're hooked on the Nintendo Wii games console.

Pensioners as old as 103 have been joining in the fun on the best-selling console, where gamers use a motion-sensitive controller to mimic sports like tennis, bowling, and boxing.

Residents have been pitting themselves against each other and games have become "competitive" since a Sunrise chef brought the console into the retirement home.

Scroll down for more...

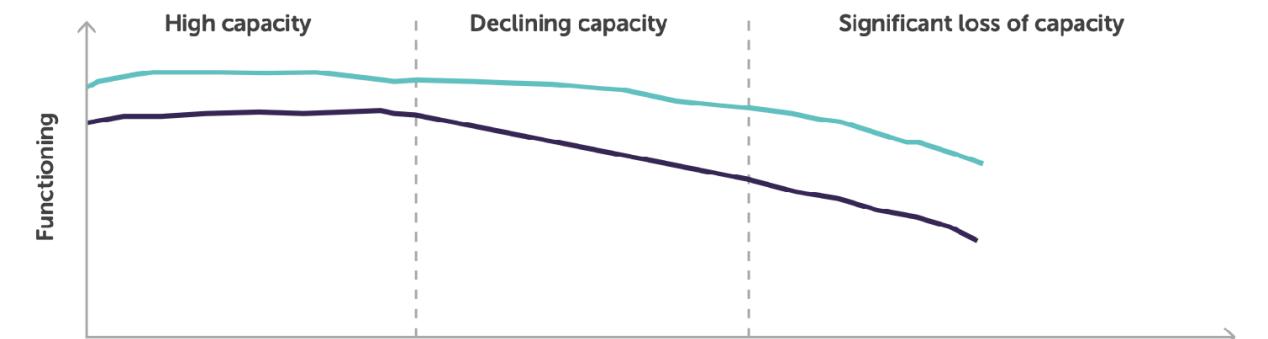


© NEWSTEAM



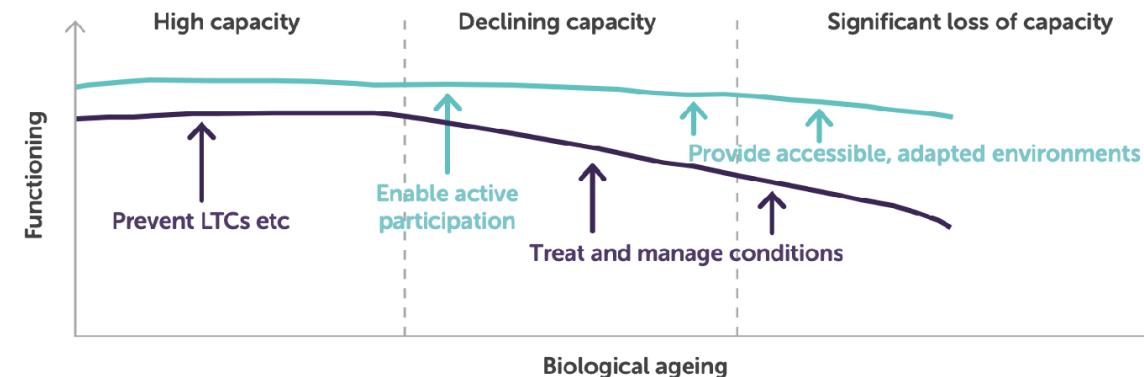
© Fabio Ciravagna, University of Sheffield

21



Additional benefits from supportive environments
— Functional ability — Intrinsic capacity

Healthy Ageing



Additional benefits from supportive environments
— Functional ability — Intrinsic capacity

<https://www.ageing-better.org.uk/publications/industrial-strategy-challenge-fund-healthy-ageing-framework>



The
University
Of
Sheffield.

- Although we have experienced a huge increase in life expectancy,
 - a significant proportion of those extra years are spent in poor health.
- In 2018 the Prime Minister announced a mission to
 - “ensure that people can enjoy at least five extra healthy, independent years of life by 2035,
 - while narrowing the gap between the experience of the richest and poorest”

<https://www.ageing-better.org.uk/publications/industrial-strategy-challenge-fund-healthy-ageing-framework>

© Fabio Ciravagna, University of Sheffield

23

Today, men aged 65 can expect to live another 19 years, but only 10 of those will be spent in good health. Women aged 65 can expect to live another 21 years, but only 11 will be spent in good health (ONS, 2017). In addition, there are significant inequalities in healthy life expectancy, with people in lower socio-economic groups developing long-term conditions at younger ages and spending a larger proportion of their later life in poor health.

There are significant opportunities for innovation across the whole spectrum of prevention, management, mitigation and adaptation. However, the private sector has been slow to respond. Despite people aged 50 and over holding an estimated 77% of the UK's financial wealth in 2014 (Centre for Economics & Business Research, 2015), there is a dearth of products and services in the market that meet the desires and aspirations, as well as the needs, of an older population.

<https://www.ageing-better.org.uk/publications/industrial-strategy-challenge-fund-healthy-ageing-framework>

© Fabio Ciravagna,

24



Need for Healthy Ageing

III-Health Impacts business and society

- ▶ Combined costs to the UK economy of worklessness and sickness absence: over £100B annually
 - ▶ £1B annually in Sheffield.
- ▶ Cost of poor mental health to local employers: £420M a year.
- ▶ 30% of the gap in productivity between North and South is attributable to the health gap.
 - ▶ 2/3 Sheffield adults are overweight or obese.
 - ▶ At least 4 in 10 are inactive
 - ▶ This impacts on cancer, CVD, MSK, etc.
- ▶ 10% reduction in long term conditions for working aged people can decrease economic inactivity by 3% (NHSA 2018)

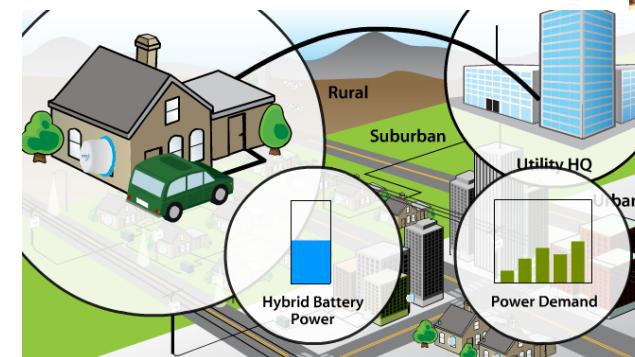
Healthy Ageing in the workforce is key

- ▶ Poor Health is the key contributor to forced early retirement (Marvell & Cox, 2017)
- ▶ An ageing UK population will escalate costs and benefits - (20m over 65s by 2066, ONS)
- ▶ Workplace Health Programmes (WHP) improve productivity but few use them
- ▶ Current solutions are not designed for 50+ users & lack 'sticky' user design & proven ROI
- ▶ Changing behaviour requires integrated solutions at every point of food choice

Source Sheffield City Council



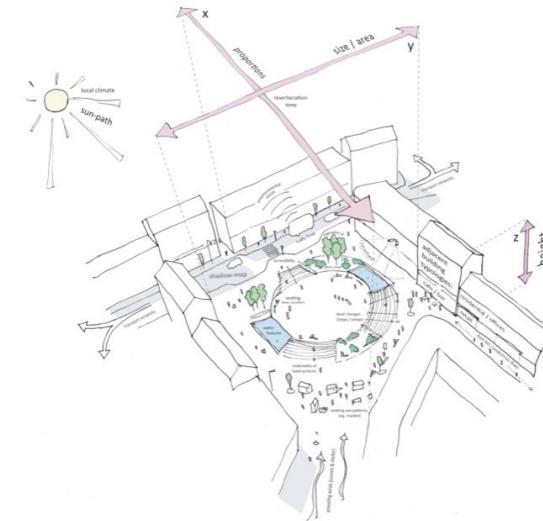
It is about Urban Spaces



© Fabio Ciravagna, University of Sheffield

26

It is about Arts and Humanities



The New York Times's success lays bare the media's disastrous state

Emily Bell



A handful of legacy institutions thrive as digital startups face increasing pressures



Humans Need Not Apply

<https://www.youtube.com/watch?v=7Pq-S557XQU>

Occupation	Number of Workers
Transportation	3,628,000
Retail salespersons	3,286,000
First line supervisors	3,132,000
Cashiers	3,109,000
Secretaries	3,082,000
Managers, all other	2,898,000
Sales representatives	2,865,000
Registered nurses	2,843,000
Elementary school teachers	2,813,000
Janitors / cleaners	2,186,000

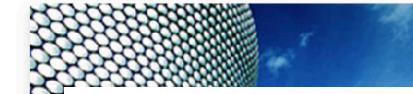
it is about commerce

Who's Gone Bust in Retail?

Who's Gone Bust in UK Retailing in 2019–2020?

Please use the links below to navigate to each section of Who's Gone Bust in retail.

- Analysis of Major Retail Failures 2008–19
- Who's Gone Bust in 2019 by Company
- What's Included and Excluded
- Archives – Pre-2019



High Street shops need to learn from online retailers if avalanche of store closures is to be reversed, experts warn

LOCATION 11 Sep 2019 by Paul Skeldon

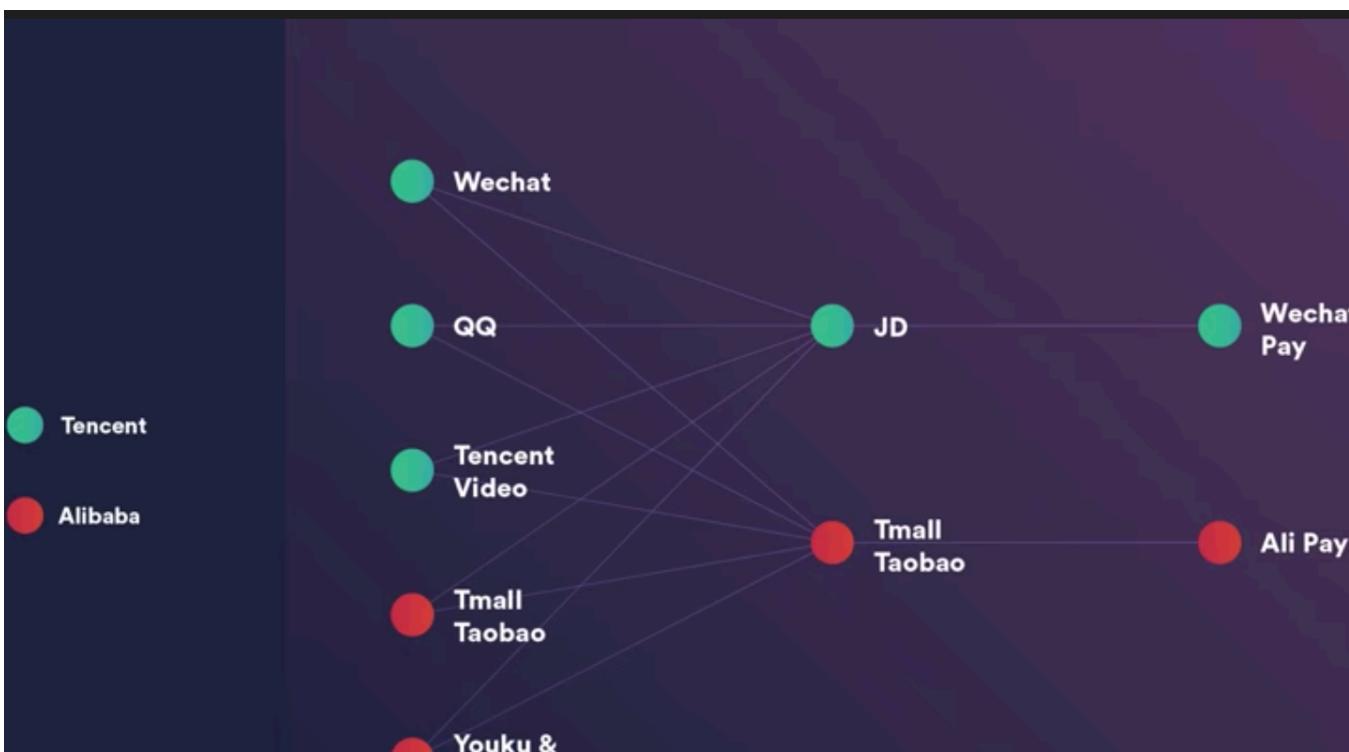
[in](#) [tw](#) [f](#) [em](#)



© Fabio Ciravagna, University of Sheffield

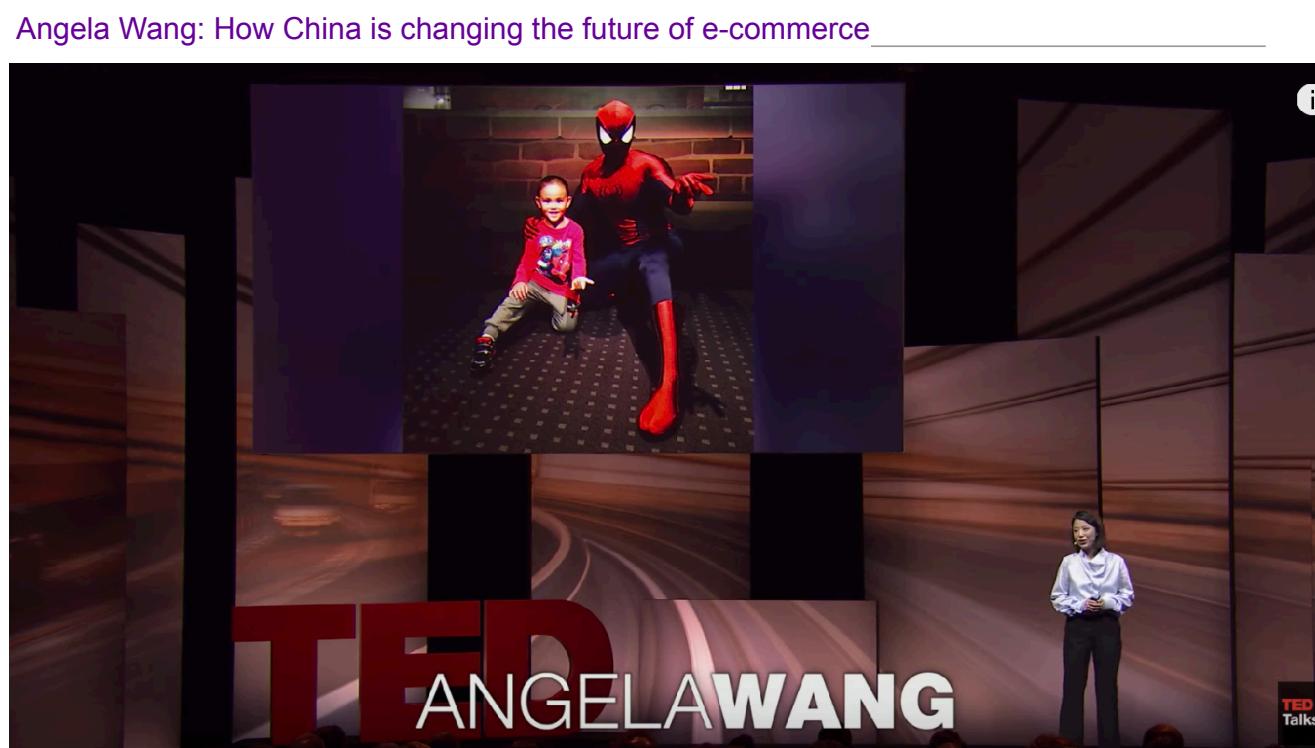
30

Integrated environment



31

It is about (e)Commerce



<https://www.youtube.com/watch?v=dOt4NkcmIUg>

© Fabio Ciravagna, University of Sheffield

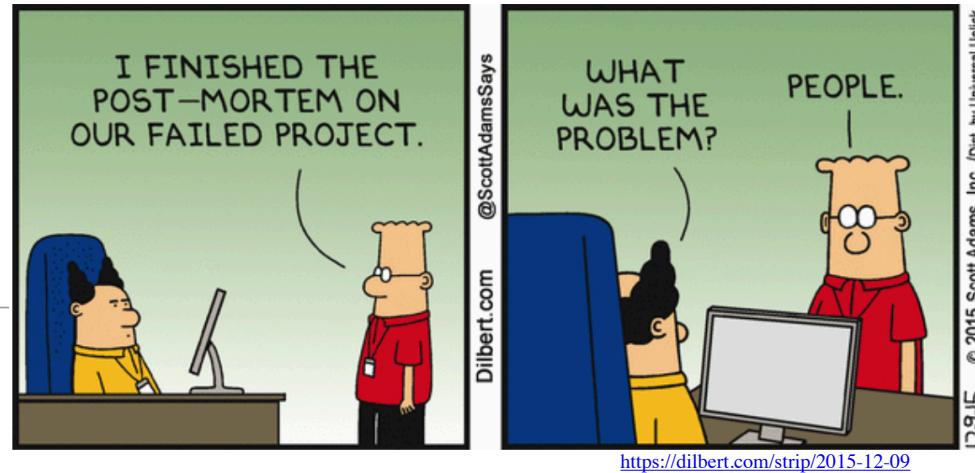
31

© Fabio Ciravagna, University of Sheffield

32



Wednesday December 09, 2015 *The Problem Is People*



Pesky Humans in the Loop

33

Internet connectivity

- Internet usage is widespread in UK and worldwide
- According to the Uk Office for National statistics
 - 87.9% of adults have used Internet in the last 3 months
- There is still an age bias
 - Looking at <45 years old the percentage increases to 98.8%
 - >75 only 38.7%
- But this will gradually disappear

Source: Office for National Statistics

<https://www.ons.gov.uk/businessindustryandtrade/itandinternetindustry/bulletins/internetusers/2016>

© Vita Lanfranchi, University of Sheffield

34

7.2.2020

<http://www.internetlivestats.com/internet-users/>

Internet Users

5,195,574,206

Internet users in the world

Around 40% of the world population has an internet connection today (view all on a page).

In 1995, it was less than 1%.

The number of internet users has increased tenfold from 1999 to 2013.

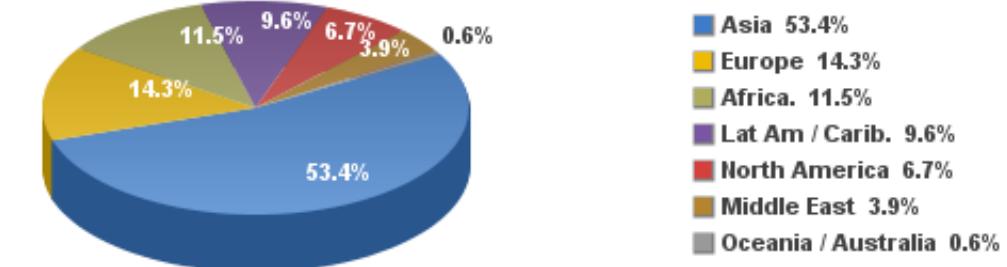
The first billion was reached in 2005. The second billion in 2010. The third billion in 2014.

© Fabio Ciravegna, University of Sheffield

35

Users by Region

Internet Users Distribution in the World - 2021



Source: Internet World Stats - www.internetworldstats.com/stats.htm

Basis: 5,168,780,607 Internet users in March 31, 2021

Copyright © 2021, Miniwatts Marketing Group

<http://www.internetworldstats.com/stats.htm>

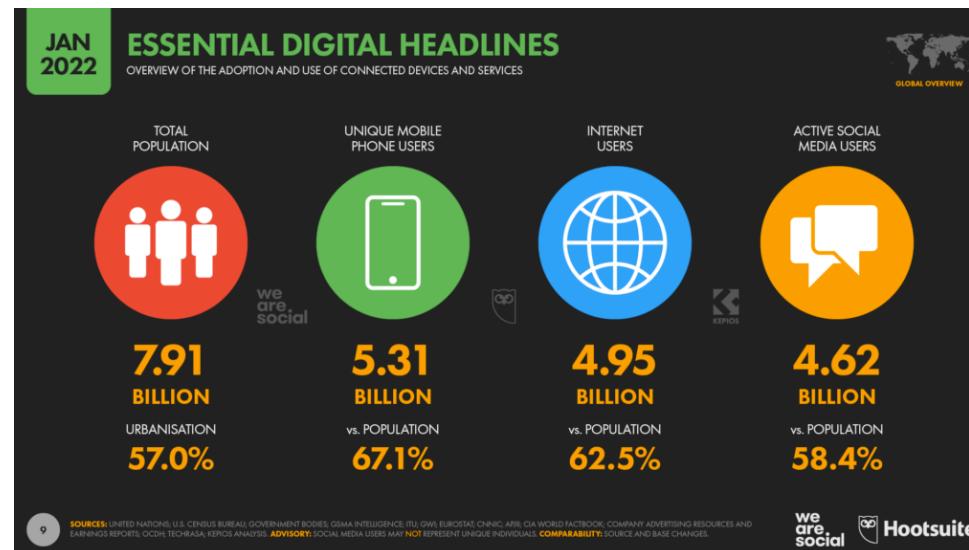
© Fabio Ciravegna, University of Sheffield

36



A look at the world

- Mobile devices are becoming the main way for people to be online
- In 2016 mobile internet access surpassed desktop access for the first time
- In 2017 mobile usage has risen of 30% whilst laptops and desktop usage has decreased of 20%



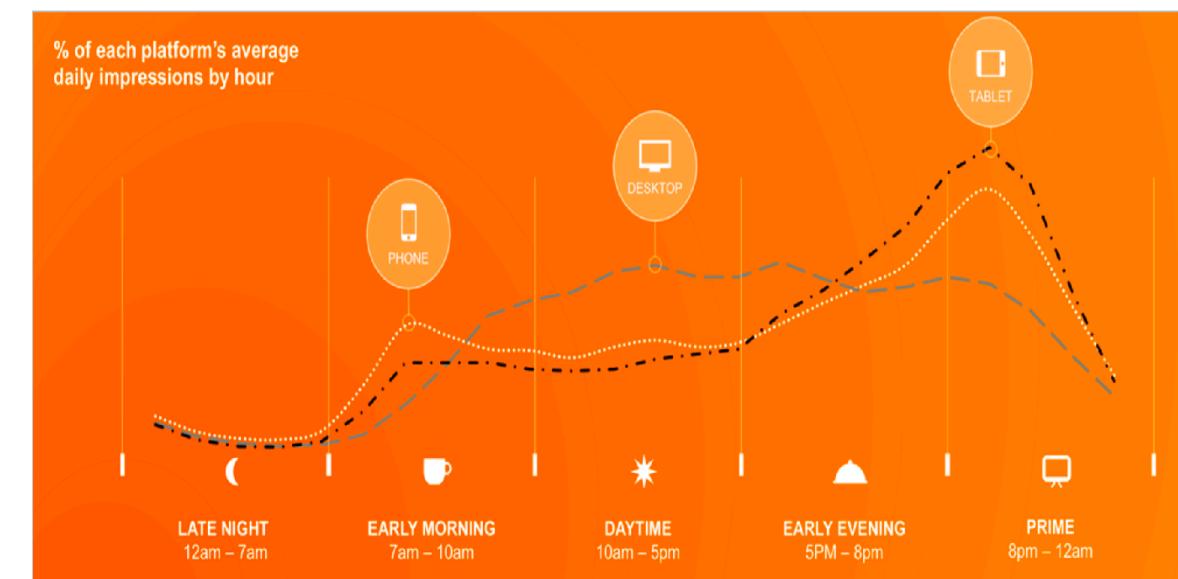
Source: WeAreSocial report <https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/>

© Vita Lanfranchi, University of Sheffield

37



Ubiquitous connection



© Vita Lanfranchi, University of Sheffield

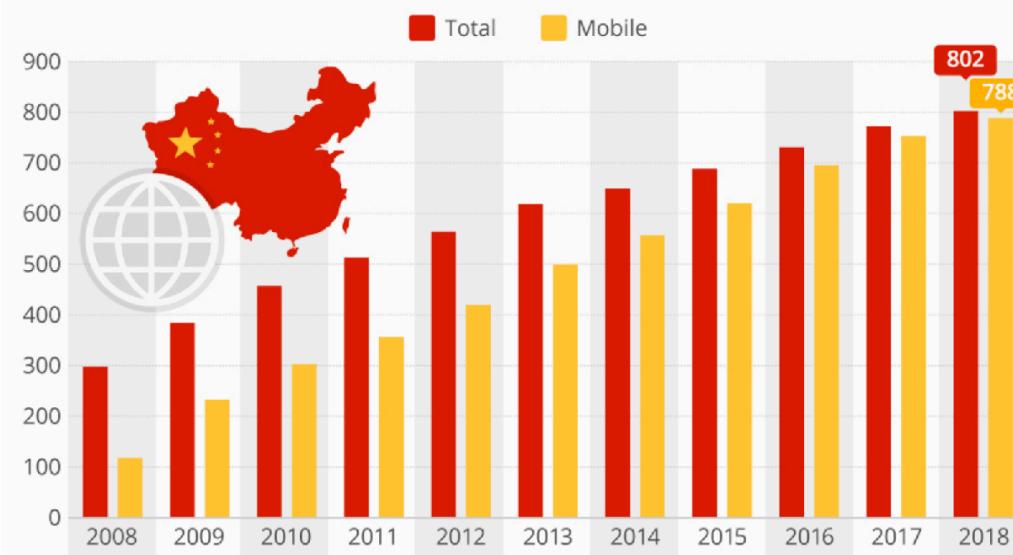
38



A worldwide outlook

98% Of Chinese Internet Users Are Mobile

Number of internet users in China (millions)



© Fabio Ciravegna, University of Sheffield

39



Mobile first

- Mobile devices are becoming the main way for people to be online

Mobile E-Commerce is up and Poised for Further Growth

Estimated mobile e-commerce sales worldwide



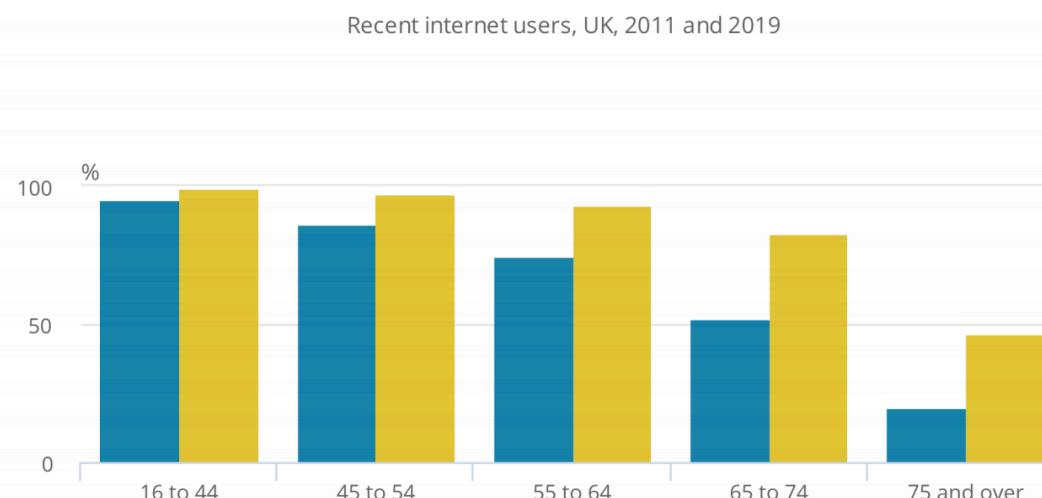
40



The
University
Of
Sheffield.

What demographics?

Figure 1: Since 2011, the 65 to 74 years age group has seen the largest increase in recent internet use



<file:///Users/fabio/Downloads/Internet%20Users,%20UK%202019.pdf>

Who are you designing your technologies for?

© Fabio Ciravegna, University of Sheffield

41

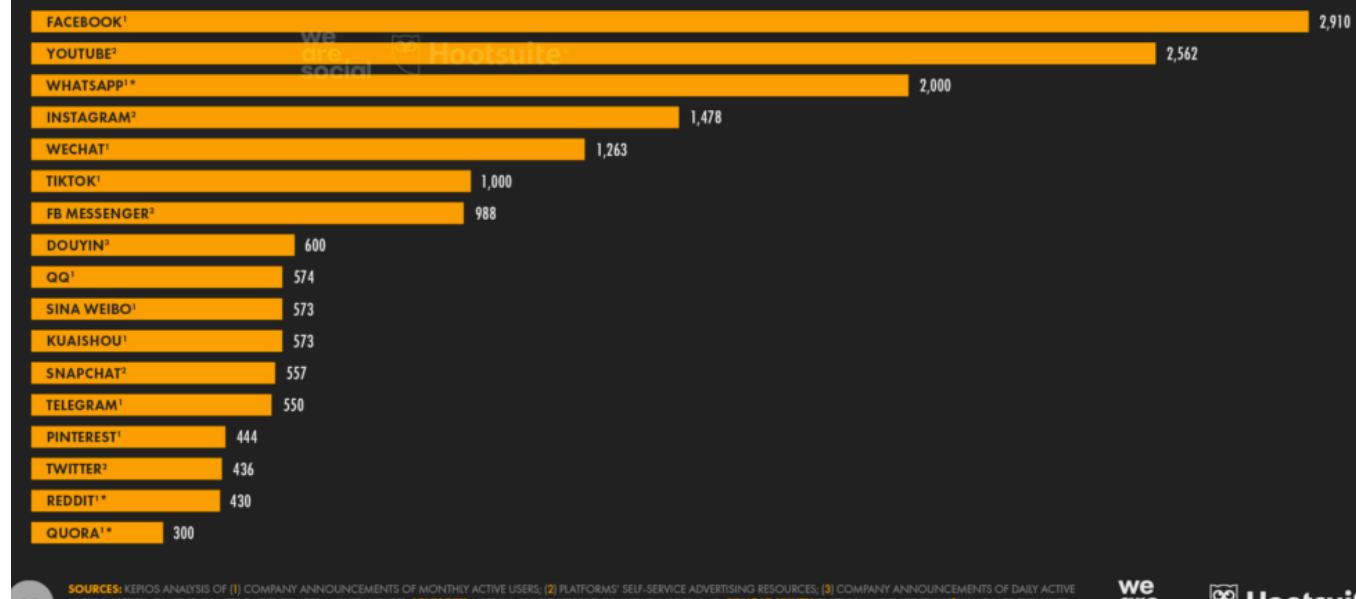


The
University
Of
Sheffield.

JAN
2022

THE WORLD'S MOST-USED SOCIAL PLATFORMS

RANKING OF SOCIAL MEDIA PLATFORMS BY GLOBAL ACTIVE USER FIGURES (IN MILLIONS)

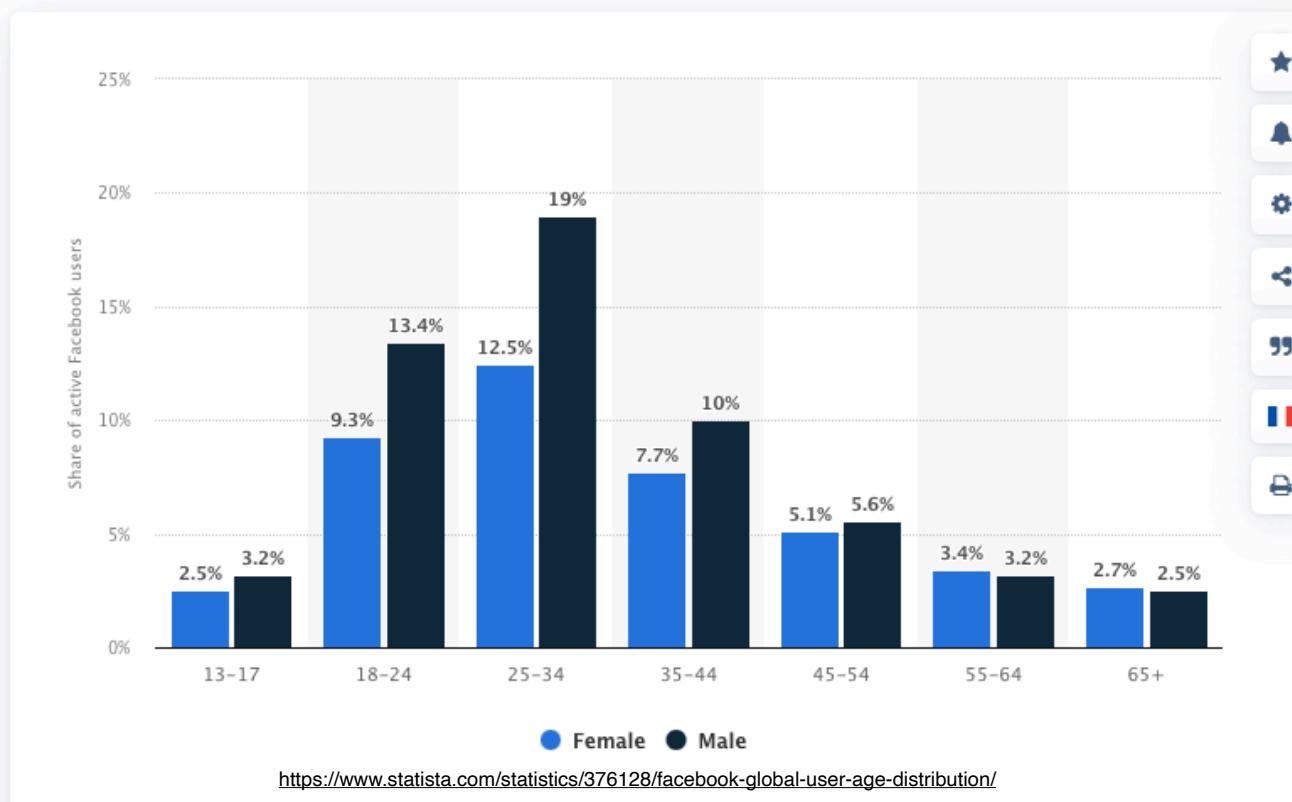


we
are
social Hootsuite

© Fabio Ciravegna, University of Sheffield

Internet > Social Media & User-Generated Content

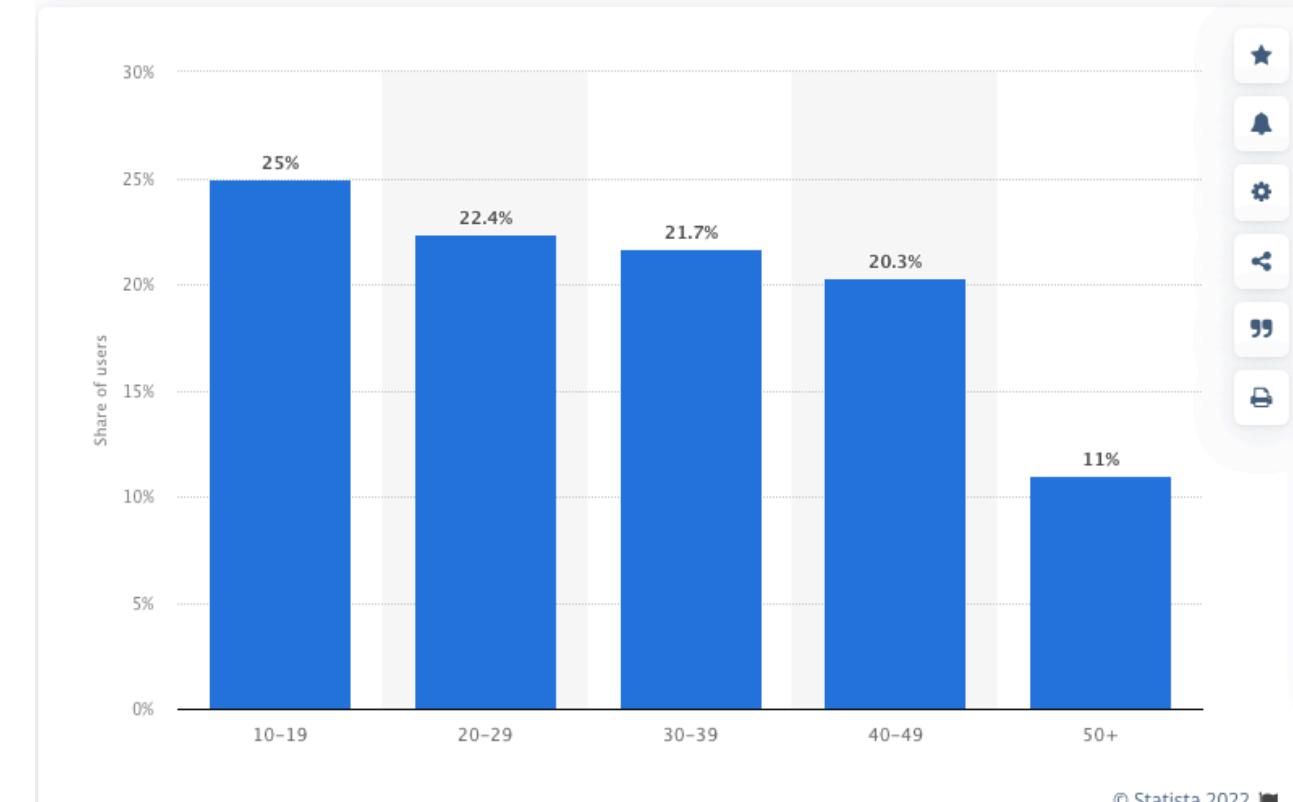
Distribution of Facebook users worldwide as of October 2021



© Fabio Ciravegna, University of Shetfield

Internet > Social Media & User-Generated Content

Distribution of TikTok users in the United States as of September 2021





https://en.wikipedia.org/wiki/List_of_countries_by_number_of_broadband_Internet_subscriptions

Broadband Penetration

Country or area	Fixed-broadband subscriptions				Mobile-cellular subscriptions			
	Number ^[7]	Rank	Subscriptions per 100 individuals ^[8]	Rank	Number ^[9]	Rank	Subscriptions per 100 individuals ^[10]	Rank
China	378,540,000	1	26.86	43	1,474,097,000	1	104.58	105
United States	109,838,000	2	33.85	24	395,881,000	4	122.01	65
Japan	40,390,640	3	31.68	28	170,128,499	7	133.45	36
Germany	33,217,000	4	40.45	10	106,000,000	15	129.09	45
Russian Federation	30,872,788	5	21.44	54	227,341,873	6	157.89	12
Brazil	28,670,016	6	13.70	74	236,488,548	5	113.00	90
France	28,429,000	7	43.75	5	69,017,000	22	106.21	99
United Kingdom	26,015,818	8	39.31	13	79,173,658	20	119.63	74
South Korea	21,195,918	9	41.58	9	63,658,688	23	124.86	55
India	17,856,024	10	1.33	126	1,186,902,277	2		128
Mexico	17,131,820	11	13.26	75	114,326,842	14		126
Italy	16,586,376	12	27.94	39	83,871,543	19	141.29	26
Spain	14,473,888	13	31.22	3				
Canada	13,922,504	14	38.01	1				
Turkey	11,924,905	15	14.77	7				
Vietnam	11,269,936	16	11.80	8				

1 South Korea is still number one – has fastest Internet speed worldwide

Posted in Mail on January 31st, 2012 by Pingdom



Want to be able to download a DVD worth of data in about 38 minutes? It may not seem very impressive, but that's with the average Internet speed in South Korea, according to the latest "State of the Internet" report by Akamai.

Covering Q3 2011, the report again puts South Korea at the top of the list of countries with the fastest Internet connections. The country scored an average connection speed of 16.7 Mbps in Q3 2011.

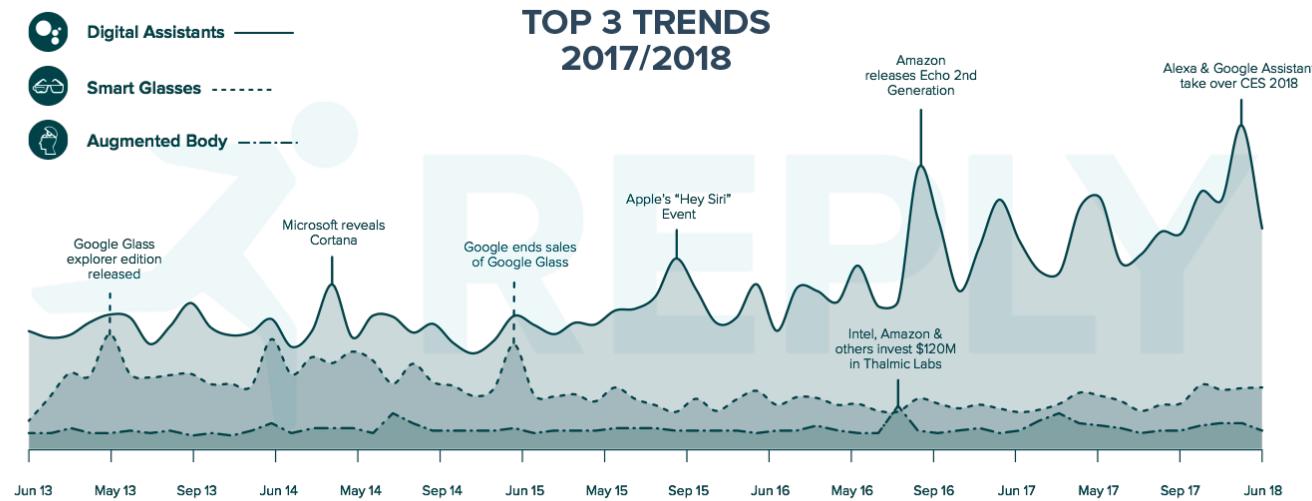
Read more

<http://royal.pingdom.com/2012/01/31/south-korea-is-still-number-one-has-fa...>

http://www.myds.com.au/img/news/NewsSuggest_GTEBM75G.jpg⁴⁶



The
University
Of
Sheffield.



<https://www.reply.com/en/topics/internet-of-things/the-evolution-of-the-consumer-internet-of-things>

© Fabio Ciravegna, University of Sheffield

47

https://en.wikipedia.org/wiki/List_of_countries_by_number_of_broadband_Internet_subscriptions



The Devices in the Loop



The
University
Of
Sheffield.

15
DEVICES
PER PERSON



... which means 15 connected devices per person.

<https://www.reply.com/en/topics/internet-of-things/the-evolution-of-the-consumer-internet-of-things>

© Fabio Ciravegna, University of Sheffield

48

...Amazon Alexa Voice Platform Goal = Faster / Easier Shopping on Amazon

Leveraging proliferation of microphones throughout house to reduce friction for making purchases...
3x faster to shop using microphone than to navigate menus in mobile apps'...

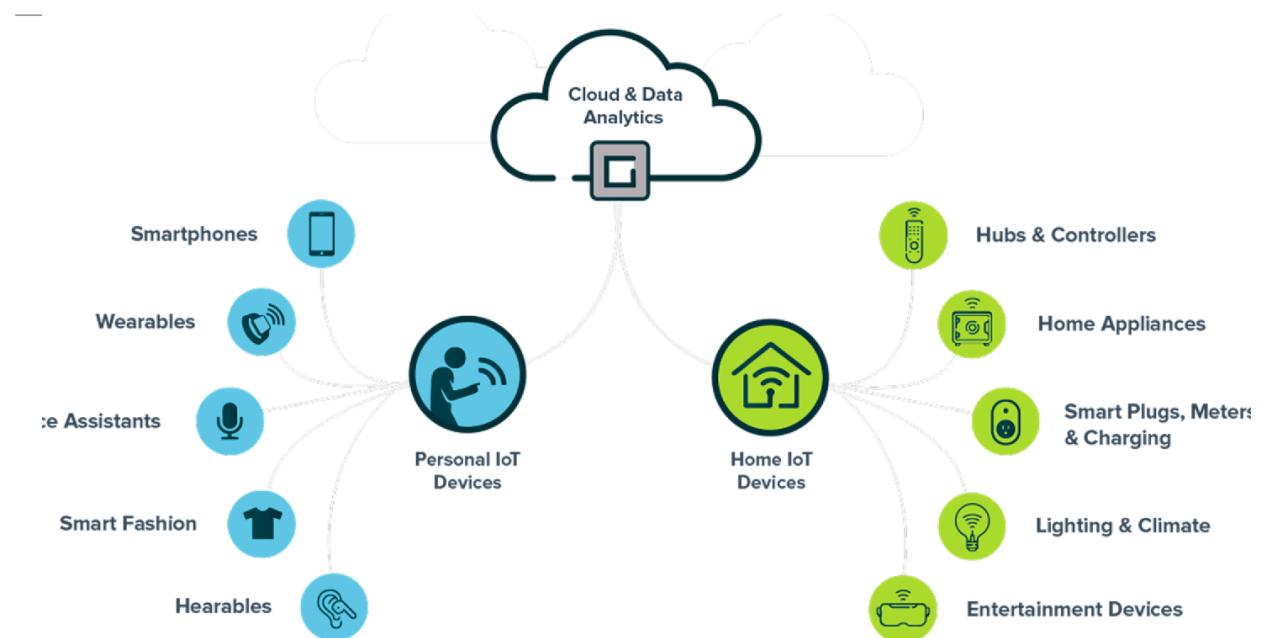


@KPCB Source: ComScore & Company Internet Panel Tracker (2014). Per capita. Millions.
Image: Amazon.com. Google Images. TechCrunch. TechCrunch.com. View full document
This slide has been shared under a Creative Commons Attribution Non-Commercial license or the following terms: 'Share alike' (Attribution share alike) under CC BY - I have not used any commercial material.

KPCB INTERNET TRENDS 2016 | PAGE 130



IoT space



<https://www.reply.com/en/topics/internet-of-things/the-evolution-of-the-consumer-internet-of-things>

© Fabio Ciravegna, University of Sheffield

50



Scannable

- Another way in into the IoT



- Access control
- Physical Access
- Cashless payments
- Targeted marketing
- Loyalty programs
- Location-based services
- Mobile payments
- In-store marketing

Some People Laugh at Wearables



Some People Laughed at PC & Internet

© Fabio Ciravegna, University of Sheffield

52

Why should everything be connected?



Technology will change retail shopping - but it's not what you think | Taylor Romero | TEDxMileHigh
<https://www.youtube.com/watch?v=TuD3IerTOms>

© Fabio Ciravegna, University of Sheffield

53

Internet Connected Buttons



© Fabio Ciravegna, University of Sheffield

55

A button for a barber shop?



© Fabio Ciravegna, University of Sheffield

56

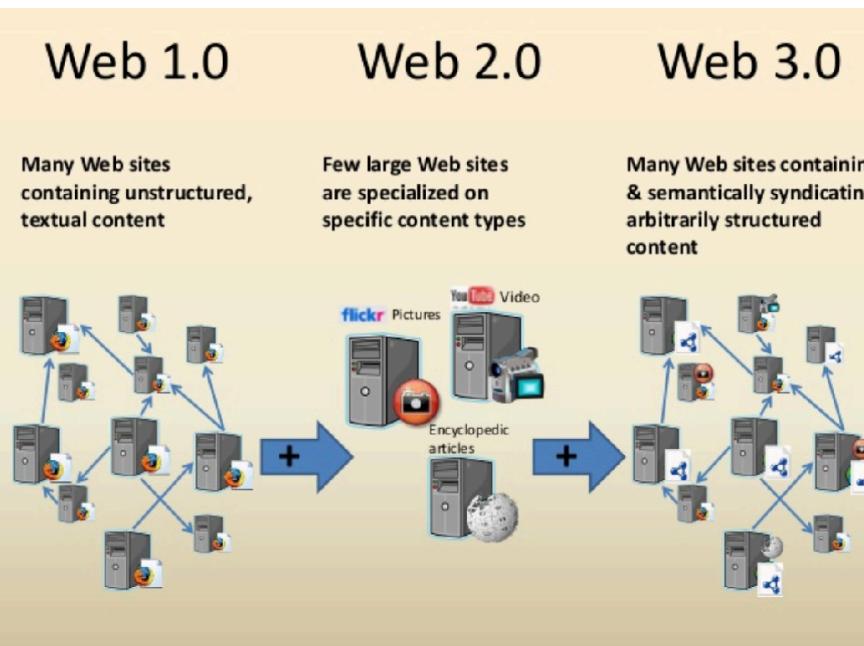
The Data

57

What is the web of data?

- Our interaction with the web has changed
 - People look for answers not for web sites
 - Websites/Apps often collect and reuse pieces of information from other sources

The evolution of the web

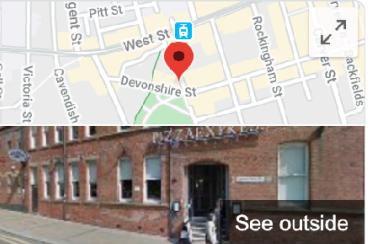


https://www.slideshare.net/soeren1611/introduction-to-the-data-web-dbpedia-and-the-lifecycle-of-linked-data?qid=2a793203-a1bf-4145-a6a4-975e2fce88b&v=&b=&from_search=4



Gooale

albert einstein



Pizza Express

[Website](#) [Directions](#) [Save](#)

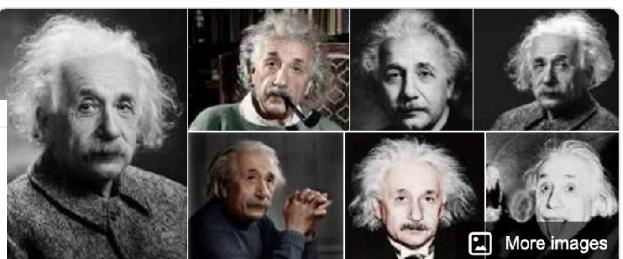
4.1 ★★★★☆ 181 Google reviews

££ · Pizza restaurant

Chain pizzeria where chefs in striped t-shirts toss handmade p
in a relaxed space.

1 min walk from Forum Kitchen + Bar

Address: 124 Devonshire St, Sheffield City Centre, Sheffield S
7SF



Albert Einstein

Theoretical physicist

Albert Einstein was a German-born theoretical physicist who
developed the theory of relativity, one of the two pillars of modern
physics. His work is also known for its influence on the philosophy of
science. [Wikipedia](#)

Born: 14 March 1879, Ulm, Germany

Died: 18 April 1955, Princeton Medical Center, New Jersey, United

Google

pizza express division

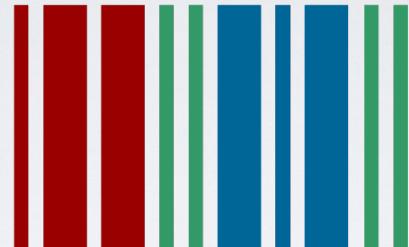
Schema.org

62

<http://www.slideshare.net/anjeve/wikidata>

Wikidata

The free knowledge base that anyone can edit



Anja Jentzsch - @anjeve

Hasso Plattner Institute, Potsdam, Germany



IT Systems Engineering | Universität Potsdam

Open Data Lecture, HTW Berlin

2015/01/12

© Fabio Ciravagna, University of Sheffield

Imagine a world in which every
single human being can freely
share in the sum of all
knowledge.

© Fabio Ciravagna, University of Sheffield

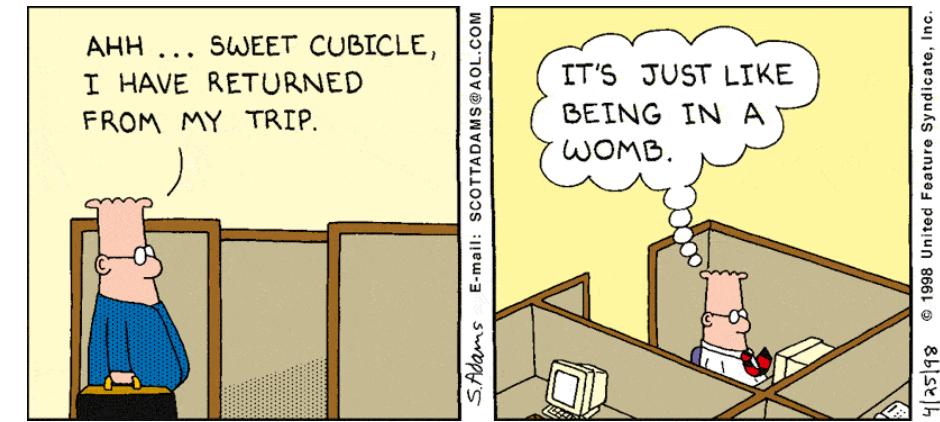


Wikidata to the rescue!

© Fabio Ciravagna, University of Sheffield



What about you?



<http://forums.sailinganarchy.com/index.php?/topic/195334-dilbert/>

If you think your job is to be a developer in your cubicle,
Think again



Questions?



The
University
Of
Sheffield.



<http://nodejs.org/>

Building a simple Client Server Architecture using HTML and Javascript

Professor Fabio Ciravegna
Department of Computer Science
University of Sheffield
f.ciravegna@shef.ac.uk
<http://staffwww.dcs.shef.ac.uk/people/F.Ciravegna/>

© Fabio Ciravegna, University of Sheffield



The
University
Of
Sheffield.

Software needed

- Download
 - Web Storm (or IntelliJ IDEA Ultimate)
 - already installed on the lab computers
 - if you use your own computers:
 - from <https://www.jetbrains.com/webstorm/>
 - You must then apply for a student licence afterwards
 - you have 30 days but do it asap
 - NodeJS (included in Webstorm - no need to download it)
 - from <https://nodejs.org/>

© Fabio Ciravegna, University of Sheffield

3



The
University
Of
Sheffield.

Learning Objectives

- In this part of the lecture we will discuss the basics of client-server communication in NodeJS
 - You will understand what nodeJS is and how it can be used to create efficient micro-services in a non blocking manner
 - The non-blocking manner is the fundamental take away message of most of the module
 - the burger restaurant analogy will haunt you for weeks, so keep it in mind!

© Fabio Ciravegna, University of Sheffield

2



The
University
Of
Sheffield.

Client Server Architecture

nodejs.org/

- In a typical client server architecture a client (e.g. a browser) communicates with a server (e.g. an Apache Tomcat server a Node server) to obtain data (e.g. a web page)
- You will have seen HTML and Javascript as a way to provide means to ask (GET) and send (POST) information to a server
- Building a server has been historically complex
- Node.js enables writing the server using Javascript
 - And it is very simple to use
 - There is a huge number of modules (libraries) that can be installed to add features and functionality – like data stores, Zip file support, Facebook login, or payment gateways.

© Fabio Ciravegna, University of Sheffield

4



The
University
Of
Sheffield.

Node.js

nodejs.org/

- Node.js is a platform built on Chrome's JavaScript runtime V8 for easily building fast, scalable network applications
- Node.js uses a model that makes it lightweight and efficient,
 - perfect for data-intensive real-time applications that run across distributed devices
- Model:
 - Event-driven
 - Non-blocking I/O
 - Built on V8

© Fabio Cravagna, University of Sheffield

5



The
University
Of
Sheffield.

Node.Js

<http://www.makeuseof.com/tag/what-is-node-js-and-why-should-i-care-web-development/>

- Event based:
 - JavaScript is an event-based language, so anything that happens on the server triggers a non-blocking event
 - Each new connection fires an event;
 - e.g. data being received from an upload form fires a data-received event
 - e.g. requesting data from the database fires an event.
 - In practice, this means a Node site will never lock up and can support tens of thousands of concurrent users

© Fabio Cravagna, University of Sheffield

6



The
University
Of
Sheffield.

Why Use Node.js?

- Performance and scalability.
 - Node is fast
- Node is perfect for offering a RESTful API
 - A web service which takes a few input parameters and passes a little data back
 - Simple data manipulation without a huge amount of computation.
 - Node can handle thousands of these concurrently where PHP would just collapse.
- It is Javascript: simple and powerful

RESTful systems typically communicate over Hypertext Transfer Protocol (HTTP) with the same HTTP verbs (GET, POST, PUT, DELETE, etc.) that web browsers use to retrieve web pages and to send data to remote servers.
REST systems interface with external systems as web resources identified by Uniform Resource Identifiers (URIs), for example /people/tom, which can be operated upon using standard verbs such as DELETE /people/tom.
https://en.wikipedia.org/wiki/Representational_state_transfer

<http://www.makeuseof.com/tag/what-is-node-js-and-why-should-i-care-web-development/>

© Fabio Cravagna, University of Sheffield

7



The
University
Of
Sheffield.

Why Node.js

<http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

- In one sentence: Node.js shines in real-time web applications employing push technology over websockets.
- What is so revolutionary about that?
 - we finally have web applications with real-time, two-way connections,
 - where both the client and server can initiate communication,
 - allowing them to exchange data freely. This is in stark contrast to the typical web response paradigm, where the client always initiates communication
 - This is a necessary requirement for many current applications including (as we will see) the very teleconferencing system we will be using for these lectures

© Fabio Cravagna, University of Sheffield

8

Drawbacks

<http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

- You definitely don't want to use Node.js for CPU-intensive operations;
- Using it for heavy computation will annul nearly all of its advantages
- Node really shines in building fast, scalable network applications,
 - as it's capable of handling a huge number of simultaneous connections with high throughput,
 - which equates to high scalability.

© Fabio Cravagna, University of Sheffield

9

Why?

<http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

- Node.js is single threaded
 - i.e. there is just one server at the till of the fast-food outlet
 - **Heavy computation could choke up Node's single thread and cause problems for all clients**
 - As incoming requests would be blocked until said computation was completed.
 - In a fast food outlet the cashier also manages the fries (and when it does it blocks the queue - imagine if he also cooked the burgers and cleaned the floor!)
- Moreover, an exception bubbling up to the core (topmost) Node.js event loop,
 - will cause the Node.js instance to terminate
 - effectively crashing the server!

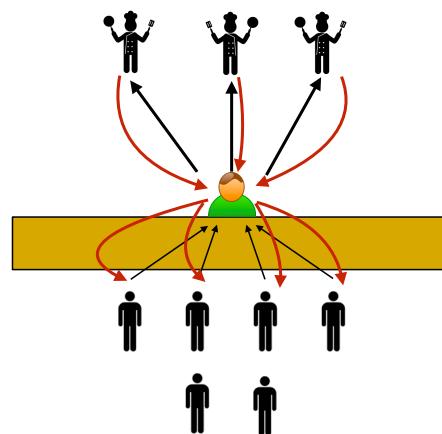
© Fabio Cravagna, University of Sheffield

10

Use non-blocking events

- Organise a node server as a burger joint

- Requests are posted to the till (the node.js server) which will direct them to the right cook (e.g. a database)
 - the till is not blocked by the time needed to prepare the food
 - While the food is being prepared, the till can serve other requests
 - When the cook has prepared the food (the data), the counter (node.js server) will return it to the client

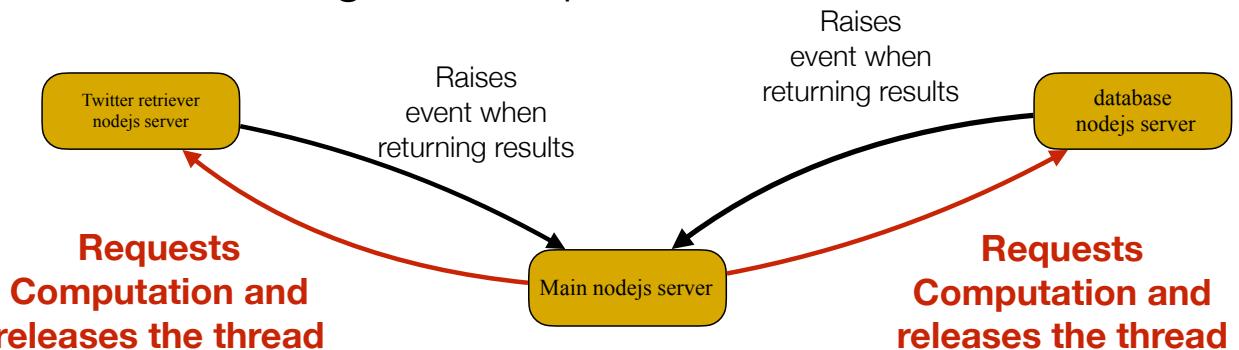


© Fabio Cravagna, University of Sheffield

11

NodeJs as a burger joint

- So organise your server so that the main loop (capturing the http/s request event) is never blocked by heavy computation
- Use a small constellation of fast specialised nodejs servers around it doing the computation



© Fabio Cravagna, University of Sheffield

12



The
University
Of
Sheffield.

NPM: node.js packages

<http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

- Package management is supported using the NPM tool that comes by default with every Node.js installation.
- NPM modules are similar to Ruby Gems:
 - a set of publicly available, reusable components, available through easy installation
 - via an online repository,
 - with version and dependency management
- A full list of packaged modules can be found on the NPM website <https://npmjs.org/>

© Fabio Cravagna, University of Sheffield

13



The
University
Of
Sheffield.

NPM Modules we will Use

- **express** - Express.js, a web development framework for Node.js, and the de-facto standard for the majority of Node.js applications out there today.
- **connect** - Connect is an extensible HTTP server framework for Node.js, providing a collection of high performance "plugins" known as middleware; serves as a base foundation for Express.
- **socket.io** and **sockjs** - Server-side component of the two most common websockets components out there today.
- **JeS** - One of the popular templating engines, a default in Express.js.
- **mongoose** - MongoDB wrappers to provide the API for MongoDB object databases in Node.js.
- **passport** - for authentication

© Fabio Cravagna, University of Sheffield

14



The
University
Of
Sheffield.

To Install an NPM package

<https://docs.npmjs.com/cli/install>

- global installation
 - installation as a global package; visible by all applications using node.js
 - **npm install <package name or url> -g**
 - This is unlikely to work on the lab computers!
 - You may not have permission to do so
- application installation
 - **npm install <package name or url>**
 - will only be recognised for the current application
 - do this on lab computers
 - this will create a **node_modules** folder under your app folder containing all the node packages installed

© Fabio Cravagna, University of Sheffield

15



The
University
Of
Sheffield.

Examples of node.js usage

<http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

- Chat is the most typical real-time, multi-user application.
 - Node.js with websockets running over the standard port 80.
 - We will see it in Lecture 6
- Api on top of a networked Database
 - Especially if lots of input is provided
 - Non-blocking operations cope with that
- Data streaming
 - We will see it in Lecture 6 (videoconference system using WebRTC)

© Fabio Cravagna, University of Sheffield

16



How to create a nodes project

- Create a nodejs project in IntelliJ
- Run it using IntelliJ
 - click run button
- Open Chrome on the local host (unless you deploy on a cloud server)
`http://localhost:<selected port>/`
 - e.g. `http://localhost:3000/`

© Fabio Cravegna, University of Sheffield

17



Thank you

© Fabio Cravegna, University of Sheffield



The
University
Of
Sheffield.

Routing and Express

Professor Fabio Ciravegna
OAK group
Department of Computer Science
University of Sheffield
<http://staffwww.dcs.shef.ac.uk/people/F.Ciravegna/>

© Fabio Ciravegna, University of Sheffield



The
University
Of
Sheffield.

What we will see

- How to create a server using Express and NodeJ
- How to declare routes in Express
 - e.g. to make sure that your server recognises a request to the address <http://localhost:3000/index>
- How to respond to a client
- Express in IntelliJ for your lab

© Fabio Ciravegna, University of Sheffield

3



Learning Objectives

- You will be introduced to Express as a way to create servers in an easy way
- You will learn
 - the spirit and syntax of Express
 - how to create routes in a server
 - i.e. URL paths the server responds to such as
 - <http://myserver.com/index> OR
 - http://myserver.com/another_route
 - how to create a project in WebStorm
 - how to access any parameter sent by a client via the body
 - how to access the data contained in the http request

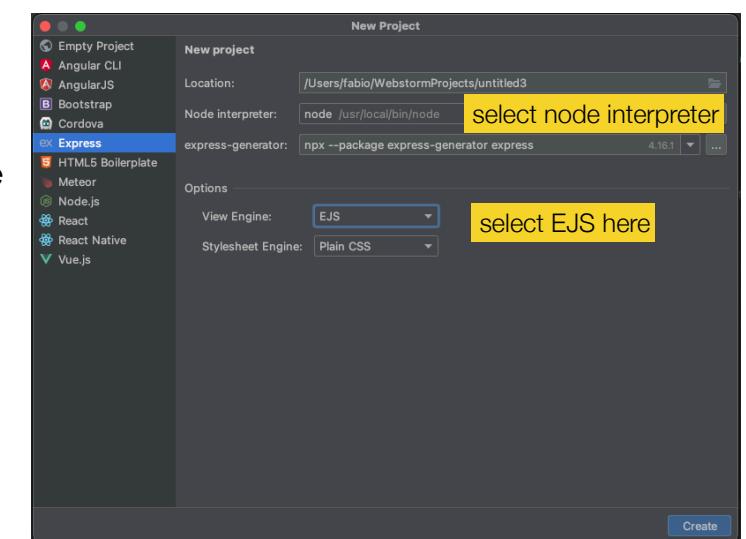
© Fabio Ciravegna, University of Sheffield

2



How to create a project in IntelliJ

- WebStorm provides an excellent plugin for node projects
- To create a project:
 - file>new>project
 - if you do not see Express then you need to install node
 - go to plugin -> install under Settings/Preference Plugins



© Fabio Ciravegna, University of Sheffield

4



The
University
Of
Sheffield.

IntelliJ: The server side in Express

routes sets in Express (a route responds to e.g. /index)

this is the Express server which internally will define two routes sets (index and users)

5



The
University
Of
Sheffield.

The server and the port

```

#!/usr/bin/env node

/**
 * Module dependencies.
 */

var app = require('../app');
var https = require('https');
var fs = require('fs-extra');

/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(val: process.env.PORT);
app.set('port', port);

/*
 * Create HTTP server.
 */

var server = http.createServer(app);

/*
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on(event: 'error', onError);
server.on(event: 'listening', onListening);

```

sets the server port

it creates the server

© Fabio Cravagna, University of Sheffield

6



The
University
Of
Sheffield.

Routing

- Routing refers to determining how an application responds to a client request to a particular endpoint,
 - which is a URI (or path) and
 - a specific HTTP request method (GET, POST, and so on)
- Each route can have one or more handler functions

© Fabio Cravagna, University of Sheffield

7



The
University
Of
Sheffield.

Nodejs: Get Routing to /

```

var http = require('http');
var url = require('url');
var server = http.createServer(function (request, response) {
  var pathname = url.parse(req.url).pathname;
  if ((pathname === '/') & (request.method === 'GET')) {
    response.end('Hello World!');
  }
});
server.listen(3000);

```

The app starts a server and listens on port 3000 for connection.
It will respond with "Hello World!" for requests to the homepage.
For every other path, it will respond with a 404 Not Found.

this is the standard node.js. We will not use it. We will use Express

© Fabio Cravagna, University of Sheffield

8



Express

<http://expressjs.com/>

- Node.js is great but most of its functions are rather verbose
- Express
 - A minimal and flexible node.js web application framework with a robust set of features for web applications
 - With a myriad of HTTP utility methods and middleware at your disposal
 - Creating a robust API is quick and easy
 - A thin layer of fundamental web application features, without obscuring Node features

© Fabio Cravagna, University of Sheffield

9

Declaring a Server In Express

```
var express = require('express')
var app = express()

app.get('/', function (req, res) {           / is a route
  res.send('I received a request on / !')
})

app.get('/hello', function (req, res) {        /hello is a route
  res.send('I received a request on /hello!')
})
```

Every time we receive a GET for “/“ or for “/hello“, then send back the appropriate string
 the client will have requested something like
 http://your.server.address:3000/hello
 we will work on a server local to your computer, so the address is
 http://localhost:3000/hello

© Fabio Cravagna, University of Sheffield

10



Routing in Express

- Route definition:

```
app.METHOD(PATH, CALLBACK)
```

- app is an instance of express retrieved using express()
- METHOD is an HTTP request method (POST, GET)
- PATH is a path on the server,
- HANDLER is the callback function executed when the route is matched

© Fabio Cravagna, University of Sheffield

11

Routing Examples

METHOD

```
// respond with "Hello World!" on the homepage
app.get('/', function (req, res) {
  res.send('Hello World!');
})

// accept POST request on the homepage
app.post('/', function (req, res) {
  res.send('Got a POST request');
})
```

PATH

CALLBACK
or
HANDLER

- app is an instance of express
- is an HTTP request method (POST, GET)
- PATH is a path on the server,
- HANDLER is the callback function executed when the route

© Fabio Cravagna, University of Sheffield

12



WebStorm

```
/* GET home page. */
router.get( path: '/index', handlers: function(req : Request<P, ResBody, ReqBody, ReqQuery, Locals>, res : Response<Re
  res.render( view: 'index', options: { title: 'My Form' });
});

router.post( path: '/index', character.getAge);

/* GET home page. */
router.get( path: '/insert', handlers: function(req : Request<P, ResBody, ReqBody, ReqQuery, Locals>, res : Response<Re
  res.render( view: 'insert', options: { title: 'My Form' });
);

router.post( path: '/insert', character.insert);
```

13

Note the helpful parameter descriptions



Route Paths

- Route paths define the endpoints at which requests can be made to.
 - e.g. '/', or '/users/' ...
- In express they can be:
 - strings
 - string patterns
 - regular expressions.
- Note!
 - Query strings are not a part of the route path.
 - In `http://localhost/index.html?index=34`
 - `?index=34` is **not** part of the route path

© Fabio Cravagna, University of Sheffield

14



Examples

```
// with match request to the root
app.get('/', function (req, res) {
  res.send('root requested')
})

// will match requests to /about
app.get('/about', function (req, res) {
  res.send('about requested')
})

// will match request to /random.html
app.get('/random.html', function (req, res) {
  res.send('random.html requested')
})
```

© Fabio Cravagna, University of Sheffield

15



String Patterns

```
// will match acd and abcd
app.get('/ab?cd', function(req, res) {
  res.send('ab?cd')
})

// will match abcd, abbcd, abbbcd, and so on
app.get('/ab+cd', function(req, res) {
  res.send('ab+cd')
})

// will match abcd, abxcd, abRABDOMcd, ab123cd, and so on
app.get('/ab*cd', function(req, res) {
  res.send('ab*cd')
})

// will match /abe and /abcde
app.get('/ab(cd)?e', function(req, res) {
  res.send('ab(cd)?e')
})
```

Note! The characters **?**, **+**, **and** ***** are subsets of their Regular Expression counterparts. The hyphen (-) and the dot (.) are interpreted literally by string-based paths. ***** means any char

© Fabio Cravagna, University of Sheffield

16



Regular Expressions

- Following the Unix standard (also used in the vim editor)

```
// will match anything with an a in the route name:
app.get('/a/', function(req, res) {
  res.send('/a/')
})

// will match butterfly, dragonfly; but not butterflyman,
// dragonfly man, and so on
app.get('.*fly$', function(req, res) {
  res.send('/.*fly$')
})
```

© Fabio Cravagna, University of Sheffield

17

Defining routes

- Route handlers for a single route path can be created using

```
app.route('/book')

  .get(function(req, res) {
    res.send('Get a random book');
  })

  .post(function(req, res) {
    res.send('Add a book');
  })
```

© Fabio Cravagna, University of Sheffield

18



app.all

- Special routing method not derived from any HTTP method
- Express will compile it into two separated requests: GET+POST with identical callback
- Used for representing all request methods.

```
// respond with "Hello World!" to all type of
// requests (post, get, etc.) on the homepage

app.all('/', function (req, res, next) {
  res.send('Got a request');
})
```

© Fabio Cravagna, University of Sheffield

19

Getting the request headers

<http://nodejs.org/api/http.html>

- Getting the request headers (e.g. user-agent)

HTTP message headers are represented by an object like this:

```
{ 'content-length': '123',
  'content-type': 'text/plain',
  'connection': 'keep-alive',
  'host': 'mysite.com',
  'accept': '*/*' }
```

- a value is requested as
 - **request.headers['user-agent']**
- this enables to access the https protocol parameters (and hence e.g. to the request metadata - e.g. to check if the request comes from a Chrome browser)
- please note!
 - **all fields are lowercase, values are not modified**

© Fabio Cravagna, University of Sheffield

20



Body-parser

- It enable accessing the parameters provided in the request

NPM modules always give you the command to use to install the module (use that on a command line)

© Fabio Cravagna, University of Sheffield

21

How to use body-parser

To respond to a form pointing to /index (see the next lab)

```
app.post('/index', function(req, res, next) {
  var login= req.body.login;
  var password= req.body.password;
```

Welcome to My Class

Please fill the form
Login:
Password:

the form fields become fields in the `request.body` field
In general, any data sent from the client will be inserted into the request's body field

We will see what `router.post` is. For now just check how to access the parameters `login` and `password` from a form

© Fabio Cravagna, University of Sheffield

22



GET request: Accessing the parameters

```
// suppose GET request, ex: http://localhost/?name=Tom

var http = require('http');

app.get('/', function(req, res) {
  var queryData = req.body;
  res.writeHead(200, {"Content-Type": "text/plain"});
  // if parameter is provided
  if (queryData.name) {
    response.end('Hello ' + queryData.name + '\n');
  } else {
    response.end("Hello World\n");
  }
});
```

© Fabio Cravagna, University of Sheffield

23

Returning results

```
result.writeHead(200, {'Content-Type': 'text/plain'});
```

- this returns code 200 (ok) to the client
- by writing it into the header of the HTTPS response

```
result.end('Hello World\n');
```

- it sends back the strings to the client
- it closes the communication

```
result.send('Hello World\n');
```

- It is equivalent to declaring `.writeHead(200) + .end(...)`

returning results must be the last instruction in `each route`. No further instructions are allowed after then.
If you forget to return the results to the client, the server hangs and stops responding to all users

© Fabio Cravagna, University of Sheffield

24



The
University
Of
Sheffield.

Defining global variables

either lasting until you turn off the server
or lasting for the duration of the specific client request

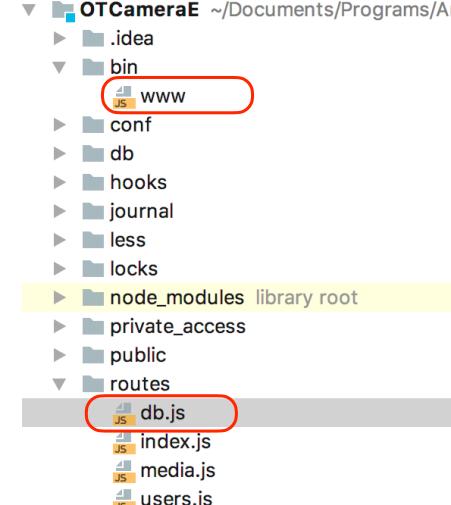
© Fabio Cravagna, University of Sheffield



The
University
Of
Sheffield.

why you want to use it?

- `app.locals` is typically used to pass data across routes and from the app to the routes



© Fabio Cravagna, University of Sheffield

27



The
University
Of
Sheffield.

app.locals

You can set and access them at any time
in the server or its routes

app.locals

The `app.locals` object is a JavaScript object, and its properties are local variables within the application.

```
app.locals.title
// => 'My App'

app.locals.email
// => 'me@myapp.com'
```

Once set, the value of `app.locals` properties persist throughout the life of the application, in contrast with `res.locals` properties that are valid only for the lifetime of the request.

You can access local variables in templates rendered within the application. This is useful for providing helper functions to templates, as well as app-level data. Note, however, that you cannot access local variables in middleware.

```
app.locals.title = 'My App';
app.locals.strftime = require('strftime');
app.locals.email = 'me@myapp.com';
```

variables persist for the lifetime of the server

they are **shared by all user requests**

locals for the lifetime of the request

or the specific user request — remember http is memoryless

© Fabio Cravagna, University of Sheffield

26



The
University
Of
Sheffield.

db route

```
router.post('/find_clients_from_remote', function (req, res) {
  console.log('find_clients_from_remote: ');
  var body = req.body;
  Database.getClient(body, function (err, dataX) {
    if (err || dataX.length == 0) {
      res.writeHead(400);
      res.end(err || 'user not found');
    } else {
      dataX = dataX[0];
      console.log('found client ' + JSON.stringify(dataX));
      var cidX = dataX._id.toString();
      console.log('cidx ' + cidX);

      // if cid is nonexistent it means it is not available
      if (isUndefined(cidX)) {
        console.log('cidx (NOT passed test) ' + cidX);
        res.writeHead(400);
        err = {status: 400, statusText: 'user not found'};
        res.end(JSON.stringify(err));
        return;
      }

      console.log('access test: ', cidX);
      var npo = req.app.locals.getNewRoom();
      console.log(JSON.stringify(npo));
      var data = {
        cid: cidX,
```

ignore this part, it is irrelevant

app locals is retrieved
using the
request object

© Fabio Cravagna, University of Sheffield

28



```
function getNewRoom() {
  var ctrlRoomId = Math.floor((Math.random() * 10000) + 1);
  var useridCitizen = Math.floor((Math.random() * 10000) + 1);
  var roomid = Math.floor((Math.random() * 100000) + 1);
  var password = Math.floor((Math.random() * 100000) + 1);
  app.locals.allowedCtrlRooms["CR" + ctrlRoomId] = "R" + roomid;
  app.locals.allowedCitRooms["CT" + useridCitizen] = "R" + roomid;
  app.locals.validPorts["R" + roomid] = "P" + password;
  return ([roomid, ctrlRoomId, useridCitizen, password]);
}

app.locals.getNewRoom= getNewRoom;
```

note: it is not assigned
using the
request object

ignore this part, it is
irrelevant

req.path

Contains the path part of the request URL.

```
// example.com/users?sort=desc
req.path
// => "/users"
```

req.protocol

The request protocol string, "http" or "https" when requested with TLS. When the "trust proxy" setting trusts the socket address, the value of the "X-Forwarded-Proto" header ("http" or "https") field will be trusted and used if present.

```
req.protocol
// => "http"
```

req.query

An object containing a property for each query string parameter in the route. If there is no query string, it is the empty object, {}.

```
// GET /search?q=tobi+ferret
req.query.q
// => "tobi ferret"

// GET /shoes?order=desc&shoe[color]=blue&shoe[type]=converse
req.query.order
// => "desc"
```



<http://expressjs.com/4x/api.html>

req.hostname

Contains the hostname from the "Host" HTTP header.

```
// Host: "example.com:3000"
req.hostname
// => "example.com"
```

req.ip

The remote IP address of the request.

If the `trust proxy` setting is enabled, it is the upstream address; see [Express behind proxies](#) for more information.

```
req.ip
// => "127.0.0.1"
```

These are used to access the HTTP request



req.accepts(types)

Checks if the specified content types are acceptable, based on the request's `Accept` HTTP header field. The method returns the best match, or if none of the specified content types is acceptable, returns `undefined` (in which case, the application should respond with 406 "Not Acceptable").

The `type` value may be a single MIME type string (such as "application/json"), an extension name such as "json", a comma-delimited list, or an array. For a list or array, the method returns the **best** match (if any).

```
// Accept: text/html
req.accepts('html');
// => "html"

// Accept: text/*, application/json
req.accepts('html');
// => "html"
req.accepts('text/html');
// => "text/html"
req.accepts('json, text');
// => "json"
req.accepts('application/json');
// => "application/json"

// Accept: text/*, application/json
req.accepts('image/png');
req.accepts('png');
// => undefined

// Accept: text/*;q=.5, application/json
req.accepts(['html', 'json']);
req.accepts('html, json');
// => "json"
```



`req.get(field)`

Returns the specified HTTP request header field (case-insensitive match). The `Referrer` and `Referer` fields are interchangeable.

```
req.get('Content-Type');
// => "text/plain"

req.get('content-type');
// => "text/plain"

req.get('Something');
// => undefined
```

Aliased as `req.header(field)`.

`req.is(type)`

Returns `true` if the incoming request's "Content-Type" HTTP header field matches the MIME type specified by the `type` parameter. Returns `false` otherwise.

```
// With Content-Type: text/html; charset=utf-8
req.is('html');
req.is('text/html');
req.is('text/*');
// => true

// When Content-Type is application/json
req.is('json');
req.is('application/json');
req.is('application/*');
// => true
```

© Fabio Cravagna, University of Sheffield

33



Getting parameters (POST)

<http://stackoverflow.com/questions/5710358/how-to-get-post-query-in-express-node-js>

.. in your route

```
// assuming POST: name=foo&color=red      <-- URL encoding
// OR POST: {"name":"foo","color":"red"}     <-- JSON encoding

app.post('/test-page', function(req, res) {
  var name = req.body.name,
      color = req.body.color;
});
```

© Fabio Cravagna, University of Sheffield

34



Response object

`res.attachment([filename])`

Sets the HTTP response `Content-Disposition` header field to "attachment". If a `filename` is given, then it sets the `Content-Type` based on the extension name via `res.type()`, and sets the `Content-Disposition` "filename=" parameter.

```
res.attachment();
// Content-Disposition: attachment

res.attachment('path/to/logo.png');
// Content-Disposition: attachment; filename="logo.png"
// Content-Type: image/png
```

© Fabio Cravagna, University of Sheffield

35



Response methods

The methods on the response object (`res`) in the following table can send a response to the client and terminate the request response cycle. If none of them is called from a route handler, the client request will be left hanging.

Method	Description
<code>res.download()</code>	Prompt a file to be downloaded.
<code>res.end()</code>	End the response process.
<code>res.json()</code>	Send a JSON response.
<code>res.jsonp()</code>	Send a JSON response with JSONP support.
<code>res.redirect()</code>	Redirect a request.
<code>res.render()</code>	Render a view template.
<code>res.send()</code>	Send a response of various types.
<code>res.sendFile</code>	Send a file as an octet stream.
<code>res.sendStatus()</code>	Set the response status code and send its string representation as the response body.

© Fabio Cravagna, University of Sheffield

36

`res.json([body])`

Sends a JSON response. This method is identical to `res.send()` with an object or array as the parameter. However, you can use it to convert other values to JSON, such as `null`, and `undefined`. (although these are technically not valid JSON).

```
res.json(null)
res.json({ user: 'tobi' })
res.status(500).json({ error: 'message' })
```



By the way Jsonp

<https://en.wikipedia.org/wiki/JSONP>

- JSONP is a trick that allows a web browser
 - to fetch JSON data from a JSON server and
 - feed them to a Javascript script in the browser.

```
<script type="application/javascript"
       src="http://server.example.com/Users/1234?callback=parseResponse">
</script>
...
<script>
  function parseResponse(param){
    ...
  }
</script>
```

© Fabio Cravagna, University of Sheffield

37



Response: redirection

`res.redirect([status,] path)`

Express passes the specified URL string as-is to the browser in the `Location` header, without any validation or manipulation, except in case of `back`. Browsers take the responsibility of deriving the intended URL from the current URL or the referring URL, and the URL specified in the `Location` header; and redirect the user accordingly.

Redirects to the URL derived from the specified `path`, with specified `HTTP status code` `status`. If you don't specify `status`, the status code defaults to "302 "Found".

```
res.redirect('/foo/bar');
res.redirect('http://example.com');
res.redirect(301, 'http://example.com');
res.redirect('../login');
```

Redirects can be a fully-qualified URL for redirecting to a different site:

```
res.redirect('http://google.com');
```

Redirects can be relative to the root of the host name. For example, if the application is on `http://example.com/admin/post/new`, the following would redirect to the URL `http://example.com/admin`:

```
res.redirect('/admin');
```

A `back` redirection redirects the request back to the `referer`, defaulting to `/` when the referer is missing.

```
res.redirect('back');
```

note! useful for login requests!!!

© Fabio Cravagna, University of Sheffield

38



`res.send([body])`

Sends the HTTP response.

The `body` parameter can be a `Buffer` object, a `String`, an object, or an `Array`. For example:

```
res.send(new Buffer('whoop'));
res.send({ some: 'json' });
res.send('<p>some html</p>');
res.status(404).send('Sorry, we cannot find that!');
res.status(500).send({ error: 'something blew up' });
```

This method performs many useful tasks for simple non-streaming responses: For example, it automatically assigns the `Content-Length` HTTP response header field (unless previously defined) and provides automatic HEAD and HTTP cache freshness support.

When the parameter is a `Buffer` object, the method sets the `Content-Type` response header field to "application/octet-stream", unless previously defined as shown below:

```
res.set('Content-Type', 'text/html');
res.send(new Buffer('<p>some html</p>'));
```

When the parameter is a `String`, the method sets the `Content-Type` to "text/html":

```
res.send('<p>some html</p>');
```

When the parameter is an `Array` or `Object`, Express responds with the JSON representation:

```
res.send({ user: 'tobi' });
res.send([1,2,3]);
```

© Fabio Cravagna, University of Sheffield

39



`res.sendStatus(statusCode)`

Set the response HTTP status code to `statusCode` and send its string representation as the response body.

```
res.sendStatus(200); // equivalent to res.status(200).send('OK')
res.sendStatus(403); // equivalent to res.status(403).send('Forbidden')
res.sendStatus(404); // equivalent to res.status(404).send('Not Found')
res.sendStatus(500); // equivalent to res.status(500).send('Internal Server Error')
```

If an unsupported status code is specified, the HTTP status is still set to `statusCode` and the string version of the code is sent as the response body.

```
res.sendStatus(2000); // equivalent to res.status(2000).send('2000')
```

© Fabio Cravagna, University of Sheffield

40



res.set(field [, value])

Sets the response's HTTP header `field` to `value`. To set multiple fields at once, pass an object as the parameter.

```
res.set('Content-Type', 'text/plain');

res.set({
  'Content-Type': 'text/plain',
  'Content-Length': '123',
  'ETag': '12345'
})
```

Aliased as `res.header(field [, value])`.

res.status(code)

Use this method to set the HTTP status for the response. It is a chainable alias of Node's `response.statusCode`.

```
res.status(403).end();
res.status(400).send('Bad Request');
res.status(404).sendFile('/absolute/path/to/404.png');
```

res.type(type)

Sets the `Content-Type` HTTP header to the MIME type as determined by `mime.lookup()` for the specified `type`. If `type` contains the "/" character, then it sets the `Content-Type` to `type`.

```
res.type('.html');           // => 'text/html'
res.type('html');           // => 'text/html'
res.type('json');           // => 'application/json'
res.type('application/json'); // => 'application/json'
res.type('png');            // => image/png:
```