

COMPSCI 326 - Web Programming

Browser-Side Storage

join on the Slack #q-and-a channel as well as Zoom

Remember, you can ask questions of your teammates on your group Slack!

please **turn on your webcam** if you can

***mute at all times** when you aren't asking a question*

(https://docs.google.com/document/d/1VA_haLv29Q5_uwCTUmXLruXt93pBvyYwCZY4_cvbM1m8/edit?usp=sharing)

Today: Browser-Side Storage

resources:

- [Storage - Web APIs | MDN](#)
- [Window.sessionStorage - Web APIs | MDN](#)
- [Window.localStorage - Web APIs | MDN](#)
- [JavaScript Cookies](#)
- [Browser storage limits and eviction criteria - Web APIs | MDN](#)

Exercise today: decoder ring - make "persistent"

- will save & restore state
- button to clear

Other stuff:

- Project ideas
- Last time: in-class exercise

Storage

Normally, when you refresh a page, it is completely cleared

However, web pages also store information locally

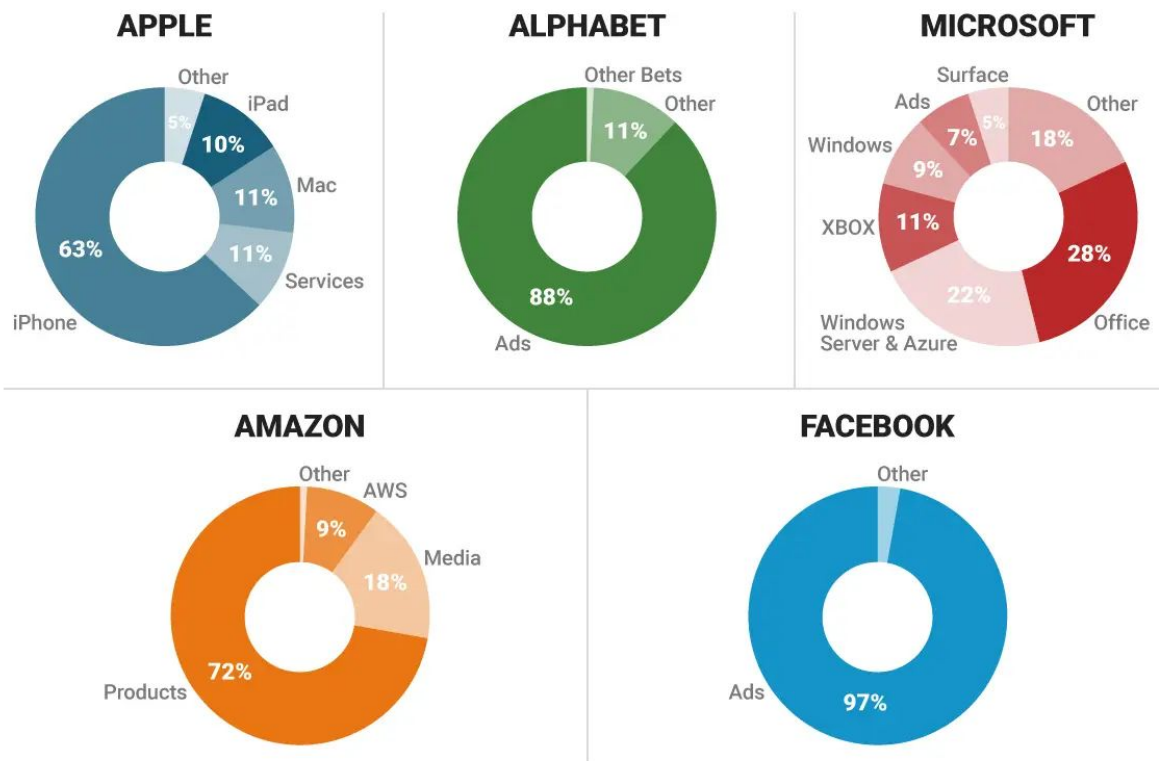
- when you last visited a site
- other data that you may already have filled in
- your username or preferences
 - **important: anyone can change local data - always need to validate on the server!**
- if you've already logged in, a session identifier
 - almost always a *hash* -- a long fixed-length string that is the result of a "one-way function" on some (usually) larger input data
 - `someOneWayFunction(data) → hash`
 - *aside*, this is used by Git to identify commits as we've seen and the familiar hash-table data structure
 - in principle, there's no way to go backwards from the hash to the data (short of brute force exploration)
 - examples: MD5 hash (<https://www.npmjs.com/package/md5>)
 - note: in general a hash is not a *cryptographic* hash (which has provable security properties)

Several kinds of storage:

- **cookies** 🍪
 - super small amount of information, returns every time you visit a page
 - document.cookie (stores a string)
 - used especially for advertising tracking
 - advertising identifiers!
 - only "the same site" can see the data stored locally for your browser, but that's not hard to do
 - cf. doubleclick = google
 - a bit about the ad ecosystem

TECH 📊 CHART OF THE DAY

REVENUE STREAMS OF THE BIG 5 TECH COMPANIES



SOURCE: Company annual reports. Figures FY2016

VISUAL CAPITALIST | BUSINESS INSIDER

(<https://www.businessinsider.com/how-google-apple-facebook-amazon-microsoft-make-money-chart-2017-5>)

- **session storage**
 - lasts until you close your browser
- **local storage**
 - lasts "forever"

- technically not true...
 - disk space limits
 - stored in your user profile on disk
 - if it gets too full, LRU policy
 - least-recently used host → all their local storage is deleted

API is key-values (storage is just an object, and **only stores strings**)

- `const storage = window.localStorage;` // or `window.sessionStorage`
- `storage.setItem("someFckingThing", "someFckingValue");` // set the key `someFckingThing` to `someFckingValue`
- `storage.getItem("someFckingThing");` // if the above was called, returns `someFckingValue`, otherwise, `null`
- `storage.removeItem("someFckingThing");` // now it's gone

Exercise!

https://docs.google.com/document/d/1y49fADRp_wEmCmxdaRDz2wUusrmUI_mnKKMwM5DfJ48/edit#