HTML 5
Local Storage

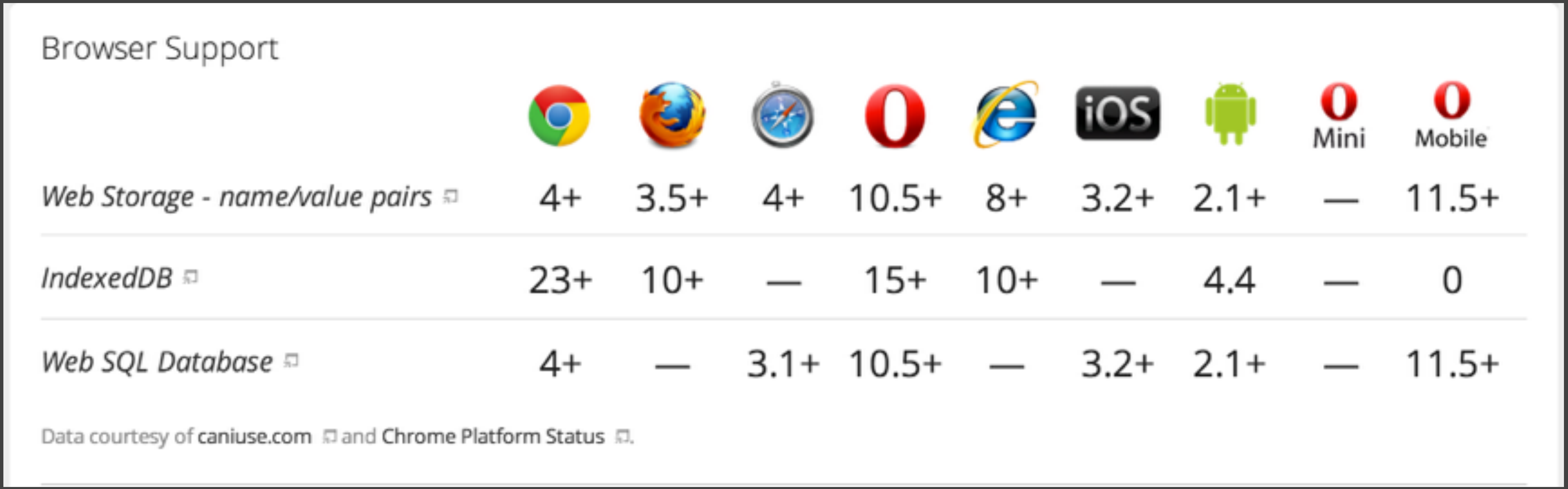
History

- First, no client side storage
- Next came cookies
  - varying cookie limits, but generally:
    - < 50 cookies per domain
    - < 4,093 bytes per domain
- Weird Java and Flash options

3 Storage Proposals

- Web Storage (aka localStorage)
  - key / value pairs
- Web SQL
  - SQL / relational storage
- Indexed DB
  - key / value pairs + indexes

Storage Support



Browser Support

						iOS		Mini	Mobile
Web Storage - name/value pairs	4+	3.5+	4+	10.5+	8+	3.2+	2.1+	—	11.5+
IndexedDB	23+	10+	—	15+	10+	—	4.4	—	0
Web SQL Database	4+	—	3.1+	10.5+	—	3.2+	2.1+	—	11.5+

Data courtesy of caniuse.com and Chrome Platform Status.

Local Storage Limits

- usually ~5MB per “origin”
  - some browsers allow up to 10MB
- no standard mechanism to request more space
- keys must be strings
- values must be strings

What's an "Origin"

- Origin = protocol + host + port
  - https://example.com/
  - http://example.com:8080/
- Also used in "Same Origin Policy"
  - JavaScript can only connect to the origin that delivered it
    - unless CORS is used (later in course)

Origin Quiz #1

Are these the same origin?

https://example.com/

http://example.com/

Origin Answer #1

Are these the same origin?

https://example.com/

http://example.com/

No - the protocol (and port) are different

Origin Quiz #2

Are these the same origin?

http://example.com/

http://www.example.com/

Origin Answer #2

Are these the same origin?

http://example.com/

http://www.example.com/

No, the hostnames are different

(even if www.example.com is a CNAME for example.com)

Origin Quiz #3

Are these the same origin?

https://example.com/

https://example.com:443/

Origin Anwer #3

Are these the same origin?

- https://example.com/
- https://example.com:443/

Yes (https defaults to port 443)

Origin Quiz #4

Are these the same origin?

http://example.com/

http://example.com:80/

Origin Answer #4

Are these the same origin?

- http://example.com/
- http://example.com:80/

Yes (http defaults to port 80)

Origin Quiz #5

Are these the same origin?

http://example.com/a/

http://example.com/b/

Origin Answer #5

Are these the same origin?

- http://example.com/a/
- http://example.com/b/

Yes, the path is not part of the origin

Key / Value Stores

- Popularized by “NoSQL” databases
  - MongoDB, REDIS, Memcache, etc.
- Keys must be unique
- Similar to
  - a 2 column database table
  - a 2 column spreadsheet

Example

Key	Value
uid	swl
displayName	Scotty Logan
mail	swl@stanford.edu
groups	staff,devs,sysadmins

localStorage API

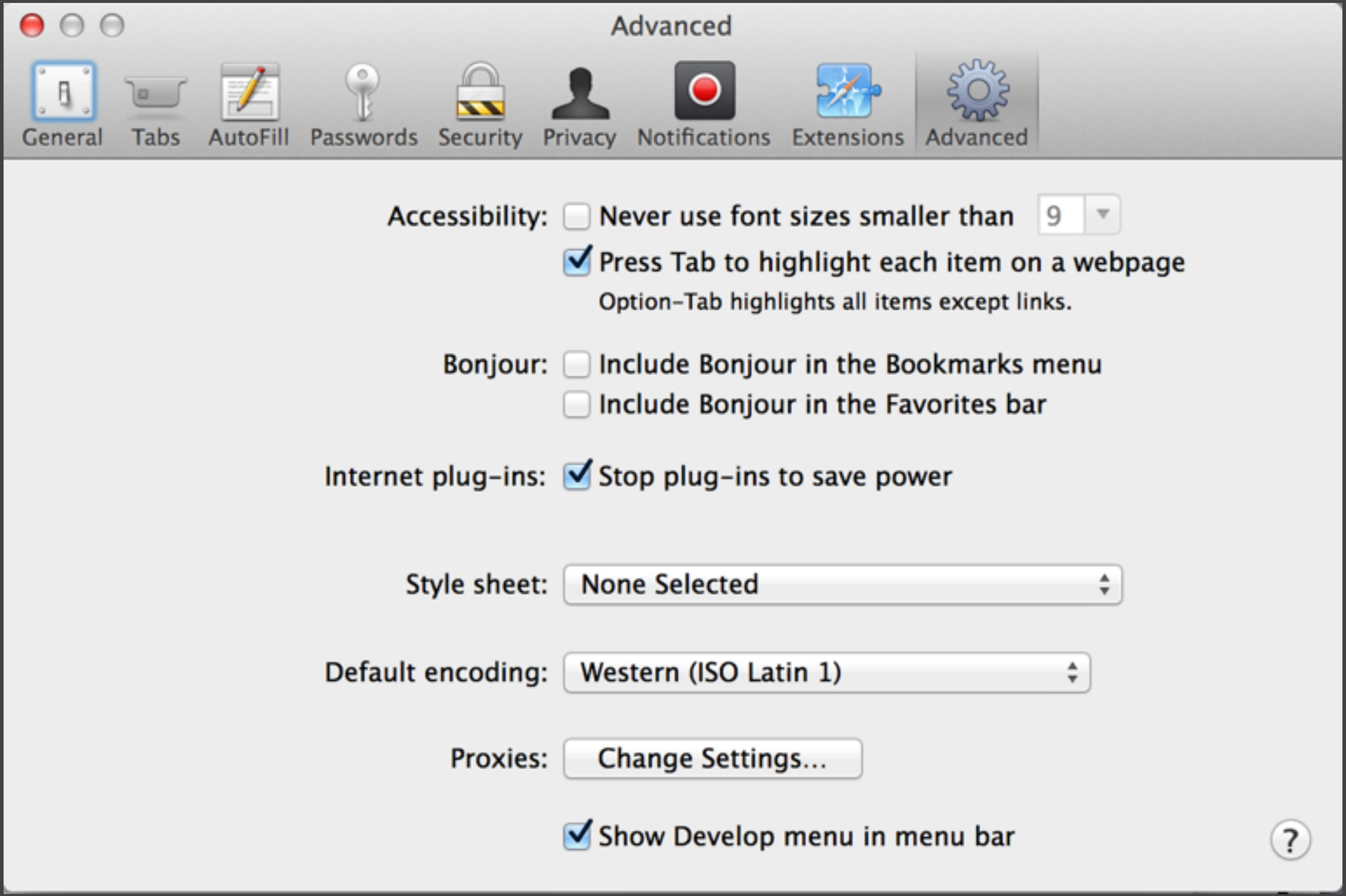
```
localStorage.length
localStorage.key(n)
localStorage.getItem(key)
localStorage.setItem(key, value)
localStorage.removeItem(key)
localStorage.clear()
```

Try it!

Open a local HTML file in your browser

Use the developer console in your browser

Safari Dev Console



Accessibility: ☐ Never use font sizes smaller than 9
✓ Press Tab to highlight each item on a webpage
Option-Tab highlights all items except links.

Bonjour: ☐ Include Bonjour in the Bookmarks menu
☐ Include Bonjour in the Favorites bar

Internet plug-ins: ✓ Stop plug-ins to save power

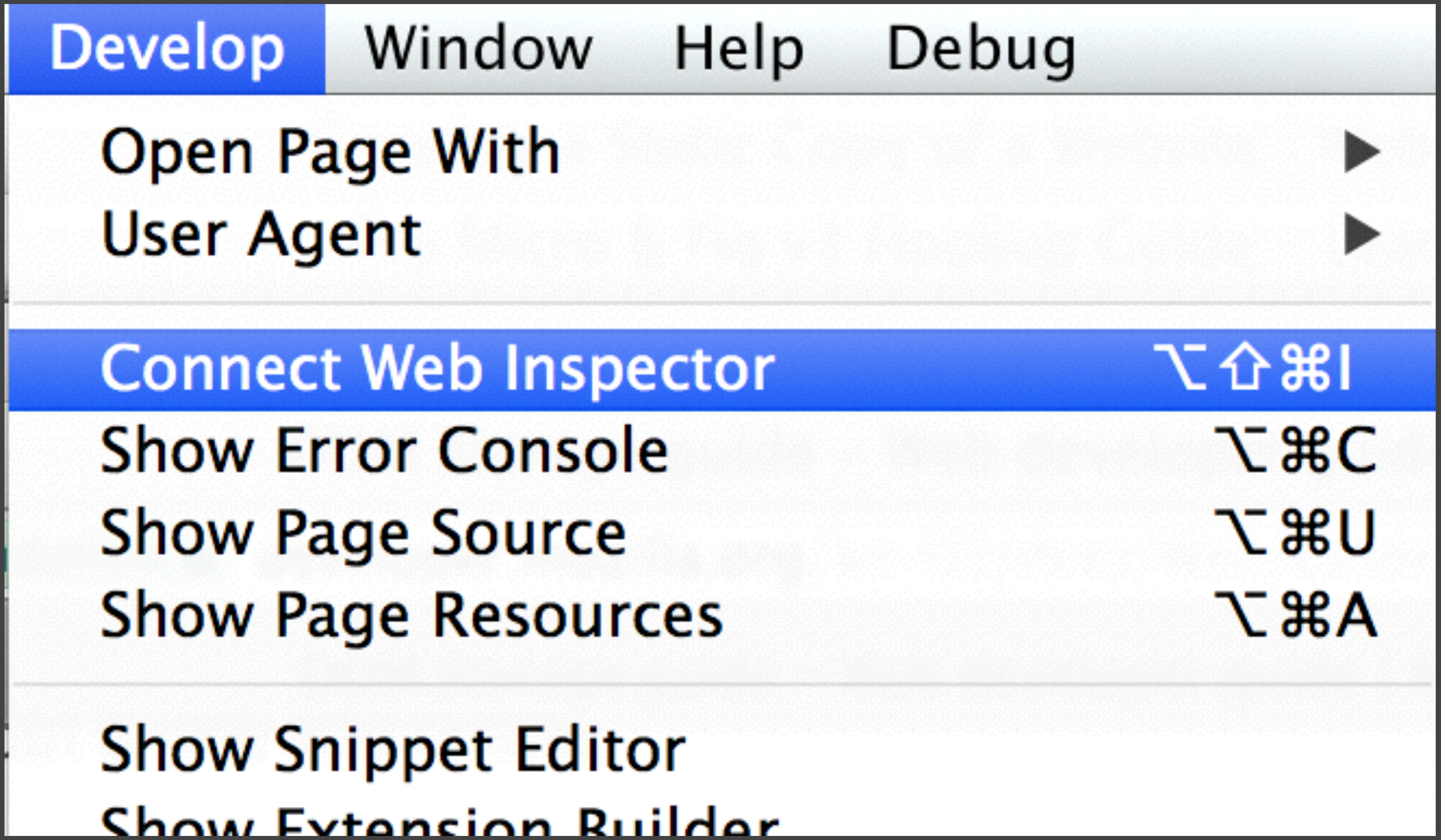
Style sheet: None Selected

Default encoding: Western (ISO Latin 1)

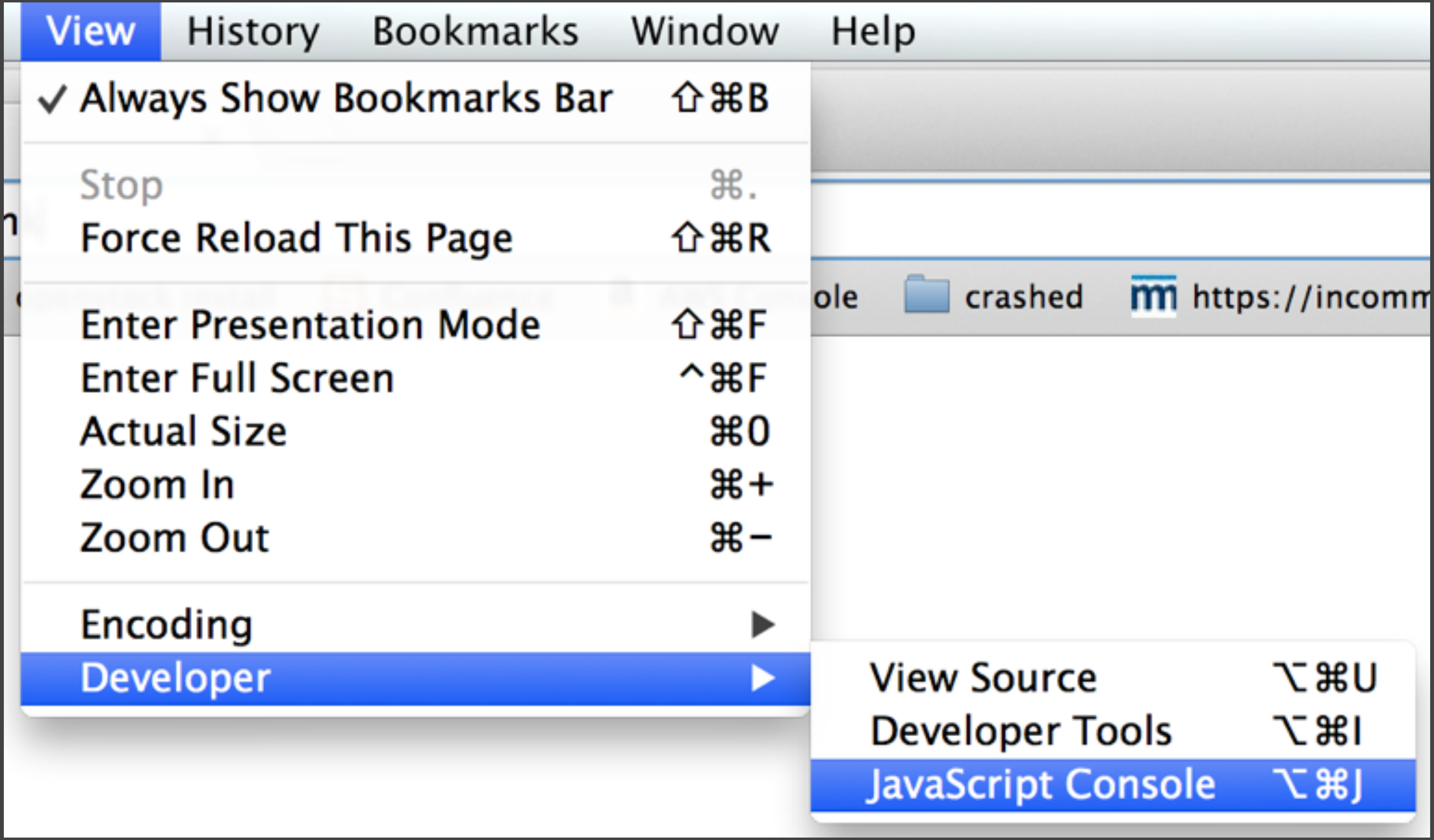
Proxies: Change Settings…

✓ Show Develop menu in menu bar

Safari Dev Console



Chrome JS Console



sessionStorage

- Same API as localStorage
- No origin policy
- Session data is just the tab or window
  - Survives reloads
  - Does not cross “open in new tab / window”

localStorage Events

- Can register for ‘storage’ events
- Each event contains 4 fields:
  - key - the key that was changed
  - oldValue - the old value, or null
  - newValue - the new value, or null
  - url - the URL that caused the change

Event Handling

```
function storageHandler (ev) {
  if (ev.oldValue === null) {
    // new key
  } else if (ev.newValue === null) {
    // key was removed
  } else {
    // value was changed
  }
}

if (window.addEventListener) {
  window.addEventListener("storage", storageHandler, false);
} else {
  window.attachEvent("onstorage", storageHandler);
}
```

Try this

```
localStorage.clear()
localStorage.setItem("test","123")
localStorage.length
localStorage.key(0)
localStorage.getItem(“test")
```

Try

setting & getting strings

setting & getting numbers

setting & getting booleans

setting & getting objects

Object Refresher

```
var obj = {
  a: 1,
  b: "Hello, World!"
}

foo({ a: 1, b: "Hello"})
```

What Did You Find?

Can you set / get strings?

Can you set / get numbers?

Can you set / get booleans?

Can you set / get objects?

Storing Objects

- Use JSON
  - Also works for numbers & booleans
- Quick version:
  - localStorage.setItem(key, JSON.stringify(value))
  - JSON.parse(localStorage.getItem(key))