

Design for the web



What we've learned so far
(hopefully!)

Web Designers just make websites.

Designers deliver information.

The **open web** enables world-wide collaboration. Your solution can make a difference.

HTML is for content. CSS is
for style.

They should work together, but never perform each other's function.

HTML should be semantic:

- Cleaner
- Accessible
- Device-agnostic
- Future-proof

CSS should be minimal,
modular and re-usable.

A person dressed in a KISS-style costume, featuring white face paint with black stripes around the eyes and a black leather outfit with silver horns. The person is looking directly at the camera with a wide, open-mouthed expression. The background is a teal-colored wall with a white, stylized floral or scrollwork pattern.

Good developers KISS.

A photograph of a desert landscape with sand dunes and camels, overlaid with a teal color filter. The scene shows a line of camels walking across the dunes, with long shadows cast across the sand. The text "Good developers DRY." is centered in white.

Good developers DRY.

A person wearing a white hoodie is sitting at a desk, leaning forward with their head buried in their hands. They are positioned in front of a laptop, which is partially visible. The background is a solid teal color. The text "Good developers refactor." is overlaid in white, centered on the image.

Good developers refactor.

When in doubt, **validate**

There are great CSS methods out there. Try them!

- SMACSS
- BEM
- OOCSS

Use frameworks that
potentialize your code.

Avoid bloat and crutches.

A site can react to viewport size in many ways.

- Fixed
- Fluid
- Adaptive
- Responsive

Code "mobile first."

Design in the order you feel
comfortable.

- Designing mobile first helps answer questions, but it's not obligatory
- Make sure it's addressed on production:
 - Start with the bare minimum – pure content.
 - Add features/style/eyecandy progressively



Consider your audience

Choose target browsers
appropriately

Lots of resources:

- [w3schools](#)
- [w3counter](#)
- [netmarketshare](#)

A cartoon illustration of a brown dog lying on its back in a desert, holding a spoon and a fork. A mechanical arm is on the left, and a winged creature is on the right. The background shows desert rock formations.

Real life: Design vs. Front-end

Always add, never block.

Show interest in each other's fields,
but respect expertise.

Example: Surveyplanet

- Developer had created entire website without interface
- Showed interest in design and openness
- We discussed every user flow/interaction
- Result: Improved ideas, better results



Communicate.

Say "I don't know" – before
it's too late.

Example: TestFlight

- Complex application for developers
- Good idea: interaction designer learn from developers
- Bad idea: know-it-all front-end developer messed up code and created problems for others

Focus on the user

- Designers too focused on their masterpiece vs. the user and the tech.
- Frond-end developers too focused on the tech vs. the user and product.

Create conventions

- Naming conventions (files, directories, layers, etc)
- Standard meetings (not too many)

MAKE GREAT STUFF!