Ma Ke

<h1>
Learning
HTML & CSS
for Web-based Apps
</h1>

Syllabus

11.30

Introduction to HTML

structure
text & media
HTML5

Hands-on construction

group i: a news on New York Times
group ii: an article on Medium blog
group iii: a tweet on Twitter

Assignment

reviewing and writing CSS codes for the constructed HTML

12.21

Introduction to CSS

property
layout
CSS3

Hands-on refinement

group i: a news on New York Times
group ii: an article on Medium blog
group iii: a tweet on Twitter

Assignment

Further elaborate the group work

12.28

Practical Web-based Apps

mobile app using HTML5 & CSS3
animation with CSS3
web-audio-based music app

Hands-on implementation

group i: apple.com mobile version
group ii: web-based animation
group iii: web-audio-based synthesizer

Assignment

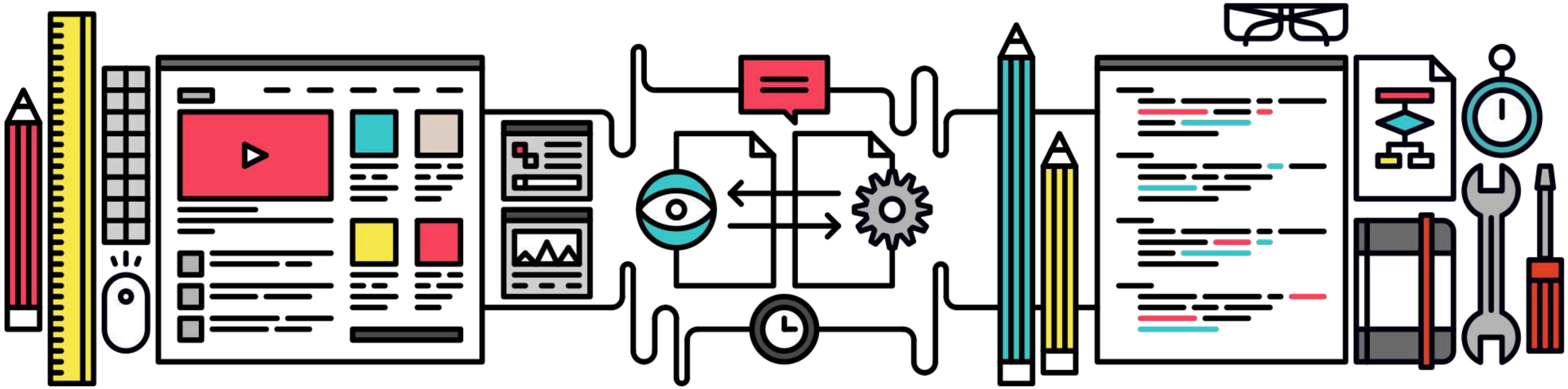
Implement the web-based apps
due to Jan 4th

<h2>
Introduction to HTML
</h2>

```
<ul>
 <li>structure</li>
 <li>text & media markup</li>
 <li>HTML5 layout</li>
</ul>
```

I ever worked here as a Front-end Developer

Ma Ke



How does Front-end Developer work for web implementation?

Group Work

Hands-on construction with HTML

Group i:
a news on New York Times

Group ii:
an article on Medium blog

Group iii:
a tweet feed on Twitter

Arrangment

Topic i: Responsible Web implementation

Topic ii: CSS Transition and Animation

Code Review

<h3>
Responsible Web Interface
</h3>

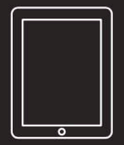
```
<ul>
 <li>screen size</li>
 <li>media query</li>
 <li>case study</li>
</ul>
```

Challenges: N versions of each web application?

SCREEN SIZES

Different visitors to your site will have different sized screens that show different amounts of information, so your design needs to be able to work on a range of different sized screens.

iPhone 4
Size: 3.5 inches
Resolution: 960 x 640 pixels



iPad 2
Size: 9.7 inches
Resolution: 1024 x 768 pixels

When designing for print, you always know the size of the piece of paper that your design will be printed on. However, when it comes to designing for the web, you are faced with the unique challenge that different users will have different sized screens.

Since computers have been sold to the public, the size of screens has been steadily increasing. This means that some people viewing your site might have 13 inch monitors while others may have 27+ inch monitors.

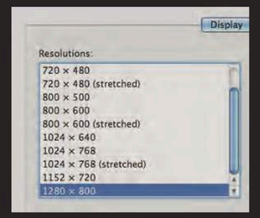
The size of a user's screen affects how big they can open their windows and how much of the page they will see. There are also an increasing number of handheld devices (mobile phones and tablets) that have smaller screens.

SCREEN RESOLUTION

Resolution refers to the number of dots a screen shows per inch. Some devices have a higher resolution than desktop computers and most operating systems allow users to adjust the resolution of their screens.

13" MacBook
Size: 13.3 inches
Resolution: 1280 x 800 pixels

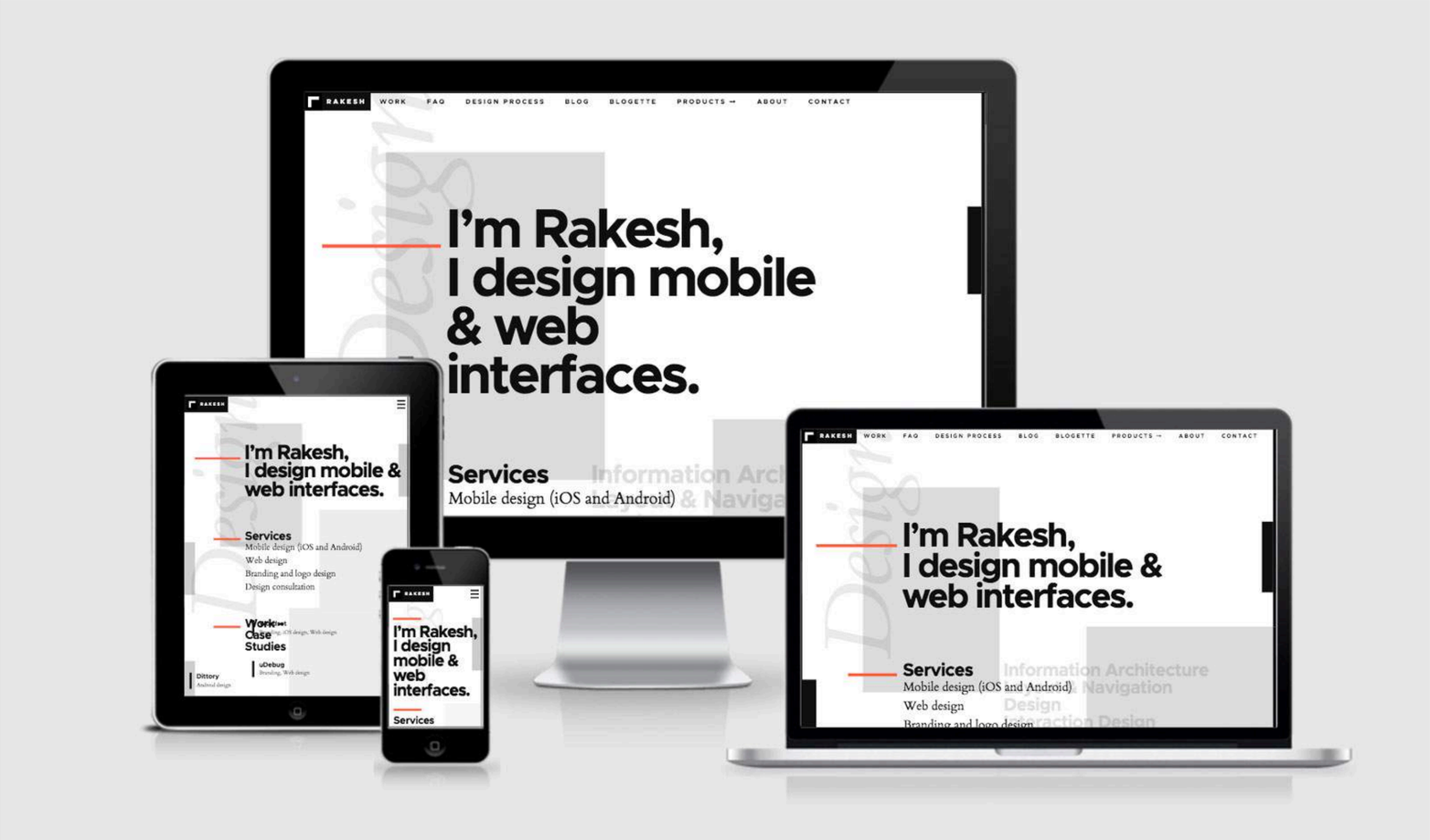
27" iMac
Size: 27 inches
Resolution: 2560 x 1440 pixels



Most computers will allow owners to adjust the resolution of the display or the number of pixels that are shown on the screen. For example, here you can see the options to change the screen size from 720 x 480 pixels up to 1280 x 800 pixels.

It is interesting to note that the higher the resolution, the smaller the text appears. Many mobile devices have screens that are higher resolution than their desktop counterparts.

Flexible solution: Responsible Layout



https://responsivedesign.is/examples/

Flexible solution: Responsible Layout

Add grid layout system with relative (e.g. 50%) rather than absolute (e.g. 50pt) measures

- Specify element packing into columns and rows

Add @media rules based on screen sizes

- Switch layout based on screen size

Making images support relative sizes

- Autoscale image and videos to fix in screen region

```
img { width: 100%; height: auto; }
video { width: 100%; height: auto; }
```

Add @media rules based on screen sizes

CSS Breakpoints

```
@media only screen and (min-width: 768px) {
/* tablets and desktop layout */
}

@media only screen and (max-width: 767px) {
/* phones */
}

@media only screen and (max-width: 767px) and (orientation: portrait) {
/* portrait phones */
}
```

Add @media rules based on screen sizes

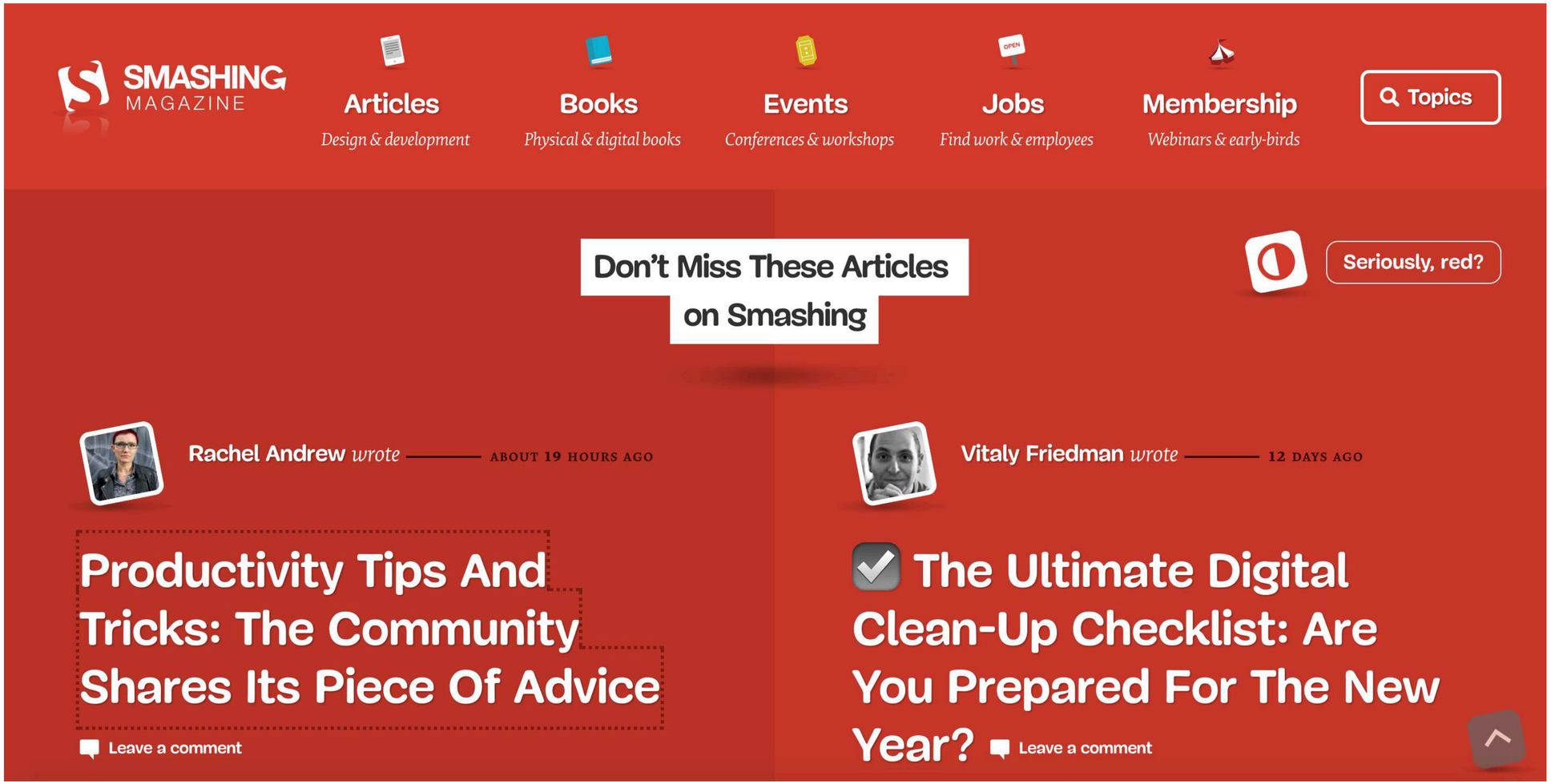
CSS Breakpoints

```
@media only screen and (max-width:1080px) {
    .work_album .grid_container{
      width: 85%;
    }
    header nav {
      width: 80%;
    }
    header nav ul li.center {
      margin: 0 10%;
    }
    section.work_album {
      padding: 8% 0 6%;
    }
}
```

Devices: PC screen

Case Studies

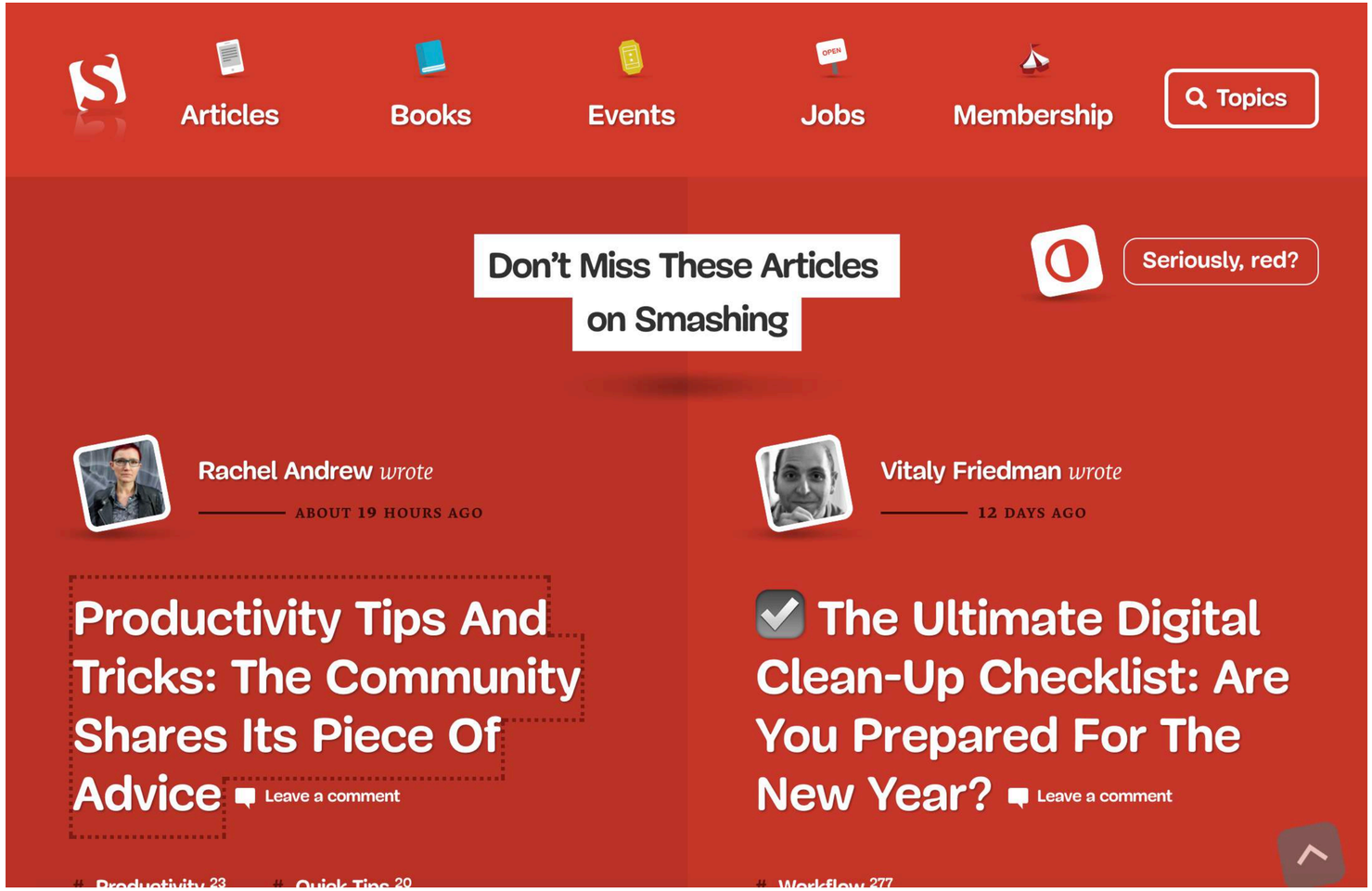
screen-width: 1440px



Devices: iPad, Tablet, Retina Phone

Case Studies

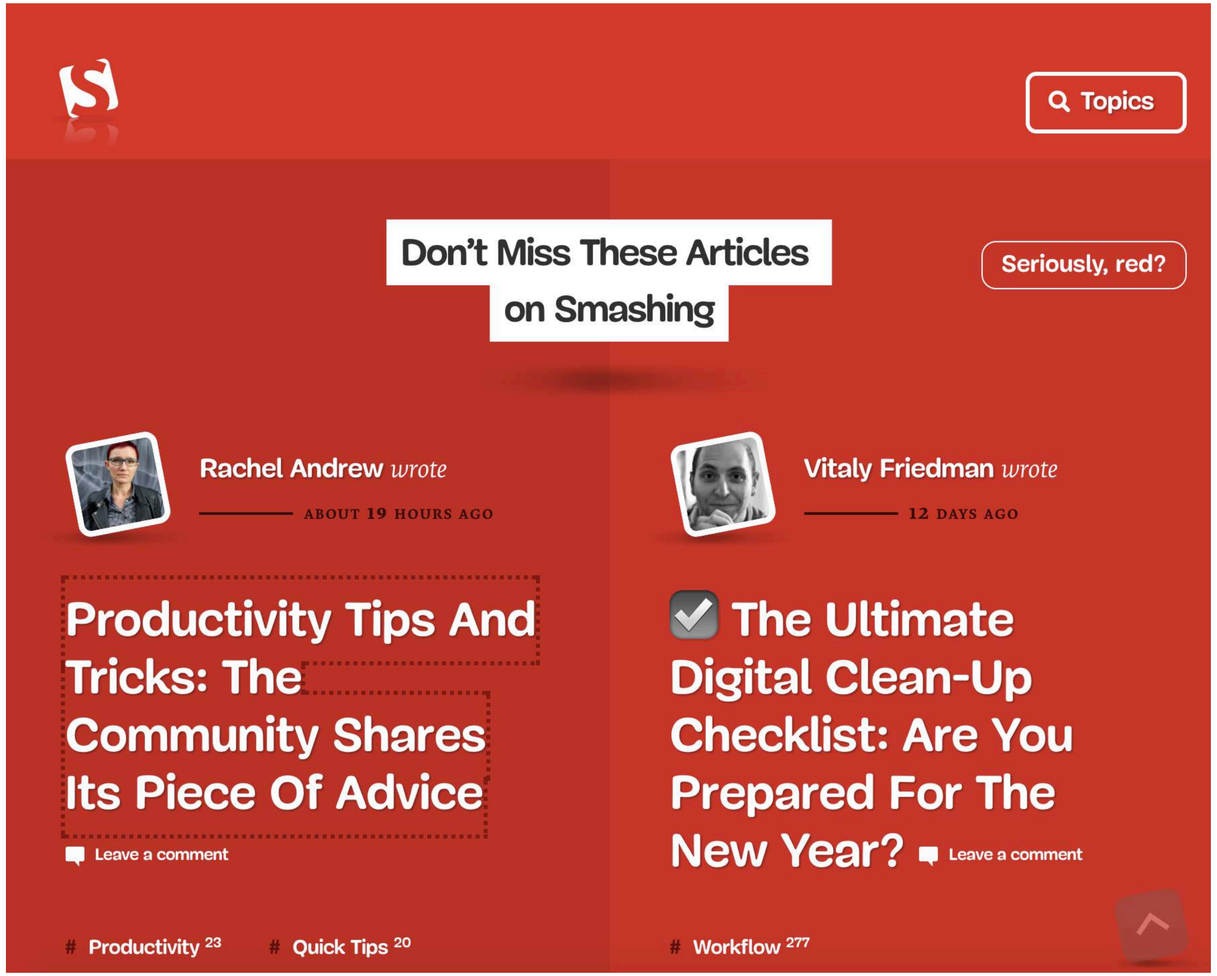
screen-width: 1100px



Case Studies

Devices: Retina Phone, iPhone Plus

screen-width: 900px



Devices: mobile Phone, iPhone

screen-width: 766px

Case Studies

Practical process

Build components to operate at different screen sizes and densities

- Use relative rather than absolute
- Specify sizes in device independent units

Use CSS breakpoints to control layout and functionality

- Layout alternatives
- App functionality conditional on available screen real estate

Mobile first popular

- Expand a good mobile design to use more real estate

<h3>
CSS Transition & Animation
</h3>

```
<ul>
 <li>Transition</li>
 <li>Animation</li>
 <li>Case studies</li>
 <li>Tools</li>
</ul>
```

- `animation-name` : declares the name of the `@keyframes` at-rule to manipulate.
- `animation-duration` : the length of time it takes for an animation to complete one cycle.
- `animation-timing-function` : establishes preset acceleration curves such as `ease` or `linear` .
- `animation-delay` : the time between the element being loaded and the start of the animation sequence.
- `animation-direction` : sets the direction of the animation after the cycle. Its default resets on each cycle.
- `animation-iteration-count` : the number of times the animation should be performed.
- `animation-fill-mode` : sets which values are applied before/after the animation.
  For example, you can set the last state of the animation to remain on screen, or you can set it to switch back to before when the animation began.
- `animation-play-state` : pause/play the animation.

`animation-timing-function`	ease, ease-out, ease-in, ease-in-out, linear, cubic-bezier(x1, y1, x2, y2) (e.g. cubic-bezier(0.5, 0.2, 0.3, 1.0))
`animation-duration`	Xs or Xms
`animation-delay`	Xs or Xms
`animation-iteration-count`	X
`animation-fill-mode`	forwards, backwards, both, none
`animation-direction`	normal, alternate
`animation-play-state`	paused, running, running

Examples

https://codepen.io/team/css-tricks/pen/ZGYZBx
https://codepen.io/anon/pen/EoWNxg
https://codepen.io/team/css-tricks/pen/c6d27fda8e1244d44bf04728049d7c1d
https://codepen.io/LabLayers/pen/MwzgNB
https://codepen.io/search/pens/?limit=all&page=20&q=css+animation
https://codepen.io/iliadraznin/pen/JcqbE
https://codepen.io/dodozhang21/pen/imIvg
https://codepen.io/Michiel/pen/EBtga
https://codepen.io/sacsooo/pen/MwxLqK
https://codepen.io/ivanodintsov/pen/LGQXmr
https://codepen.io/captainbrosset/pen/WbzGRw
https://codepen.io/digitalcraft/pen/LEryJK

Tools

animista

http://animista.net/play/basic