

# Responsive Web Design

## Core Theory



기술을 이용해, 동료를 이용해, 세상을 이용해

Responsive Web Design  
**RWD** 웹사이트 디자인

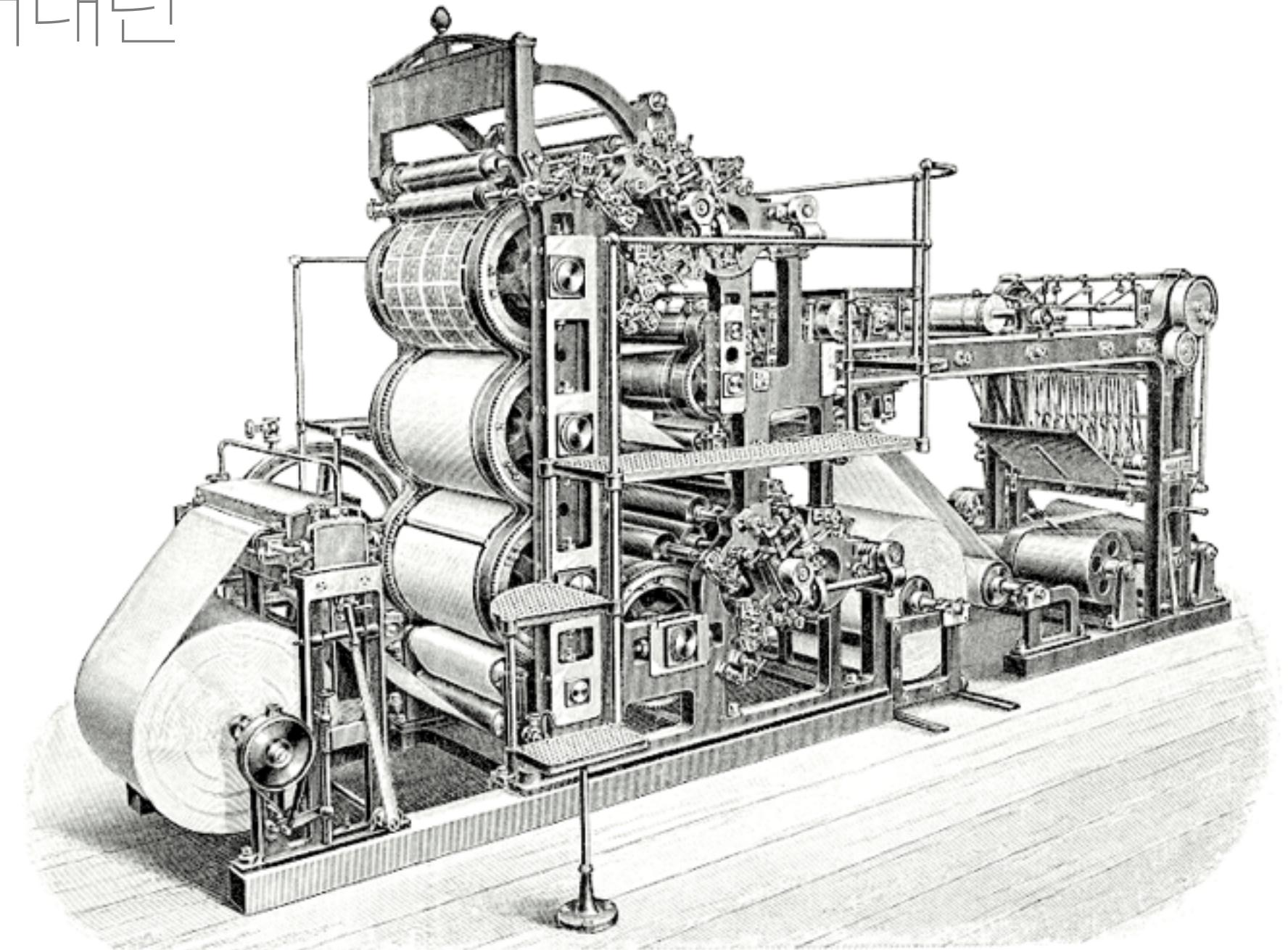
이롭게

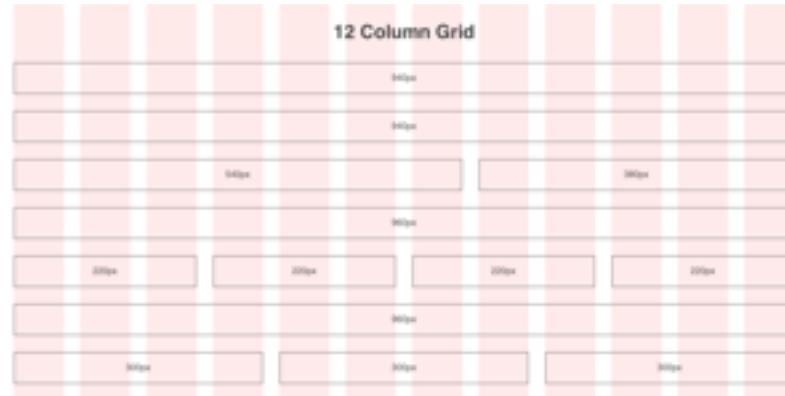
# 고정된 판형을 짹어내던 시대는 끝났다.

It's over.

Waterfall model doesn't make that much sense in the responsive age. That model worked 3/4 years ago when we lived in a different context with different devices.

기술을 이용해, 동료를 이용해, 세상을 이용해



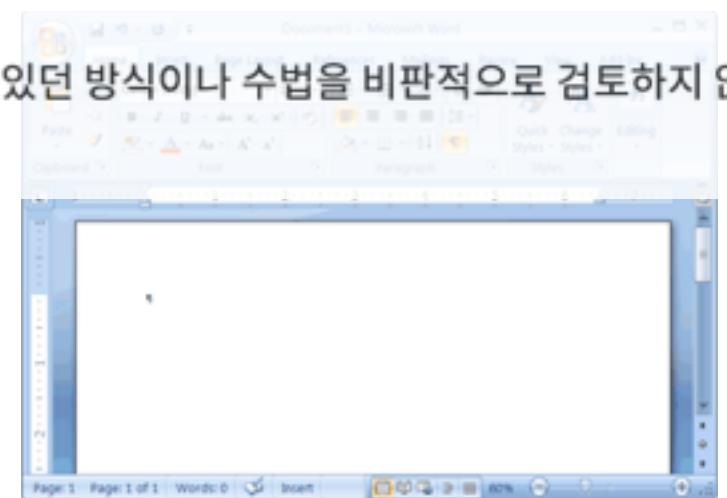


# 과거를 답습하는 것은 이제 그만!

Desktop app / fixed grid / Mobile interface

답습 踏襲

전부터 해 내려오거나 있던 방식이나 수법을 비판적으로 검토하지 않고, 있는 그대로 받아들  
이거나 따름





오늘을 연구하고 만들어야 한다.

Smart TV / Web App / Mobile / Console / Responsive design

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



# 브라우저 전쟁 VS 테크놀로지의 발전

Browser war VS Technology evolution

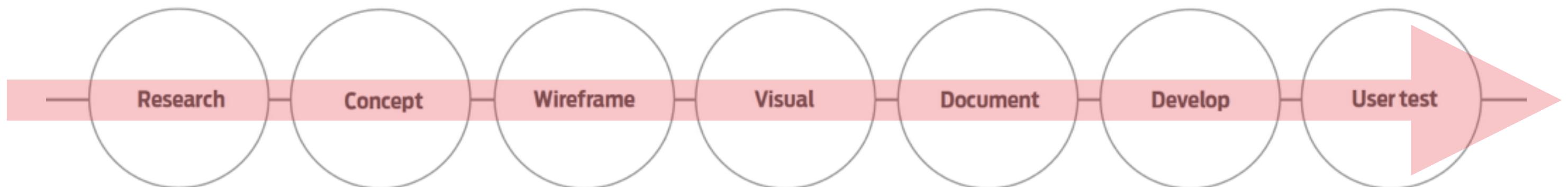
기술을 이용해, 동료를 이용해, 세상을 이용해



# 전형적인 UX/UI 디자인 프로세스

시장 분석, 컨셉 도출 부터 시장에 제품을 출하하기 까지 오직 직선만 그릴 뿐인 과거의 방법 (직렬)

기술을 이용해, 동료를 이용해, 세상을 이용해

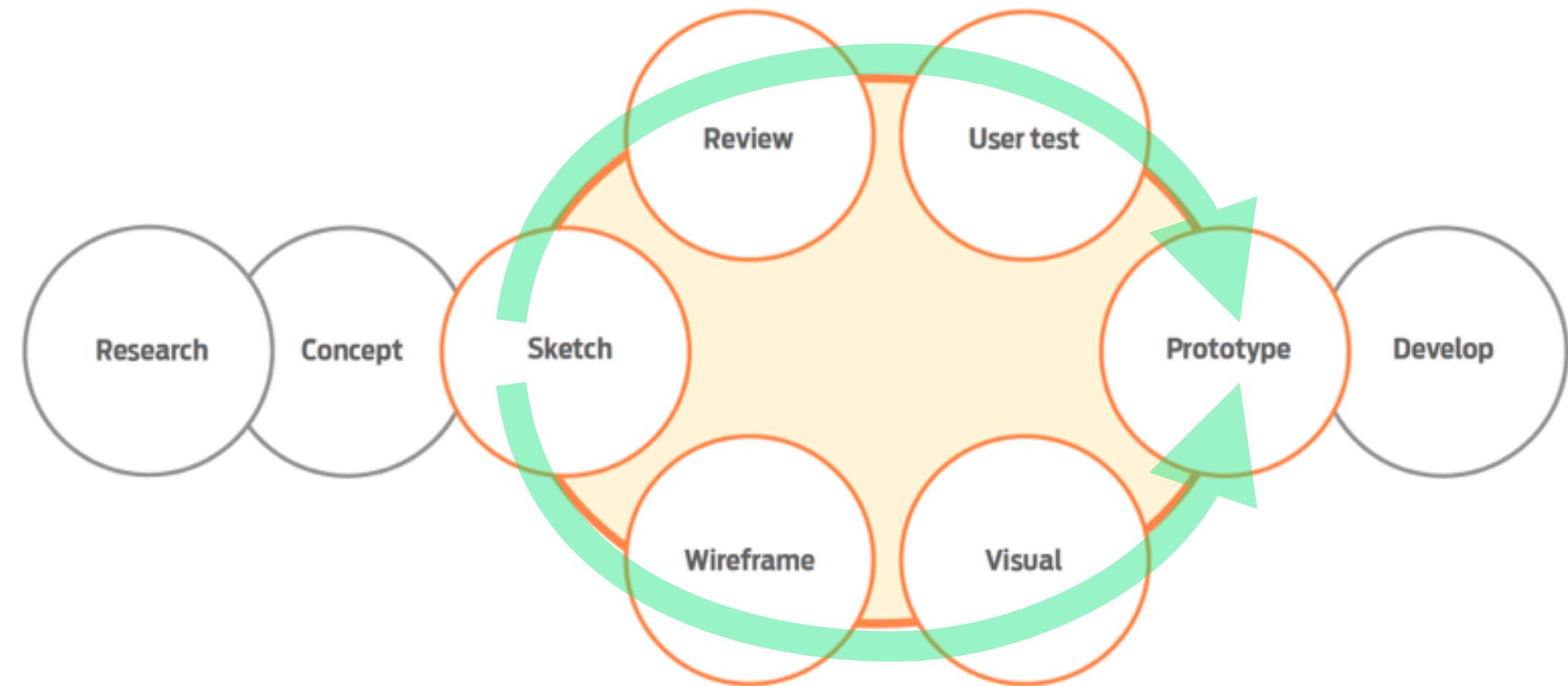


IROPKE Front-End Design CAMPUS

# 새로운 UX/UI 디자인 프로세스

시대가 변한 것처럼, 디자인 프로세스도 변해야 한다 (병렬)

기술을 이용해, 동료를 이용해, 세상을 이용해



기술을 이용해, 동료를 이용해, 세상을 이용해





기술을 이용해, 동료를 이용해, 세상을 이용해  
웹을 이용할 수 있는 **멀티 디바이스**  
그것을 이야기하고자 하는 것이 아니다.

기술을 이용해, 동료를 이용해, 세상을 이용해  
우리가 만들어야 할 서비스를 이루는  
콘텐츠와 컨텍스트에 관한 이야기이다.

# 테크놀로지의 발전은 디자인 바운더리를 변경시켰다.

The evolution of technology is changing the boundaries of design

기술을 이해하기, 기술을 활용하기, 기술을 적용하기, 기술을 활용하기



Digital Experiences

동일한 콘텐츠  
다른 모양에 담을 뿐.

Same contents, different shapes

기술을 이용해, 독창성을 이용해, 혁신을 이용해

Extended Fluid Experience



# 콘텐츠는 물과 같다.



기술을 이용해, 동료를 이용해, 세상을 이용해

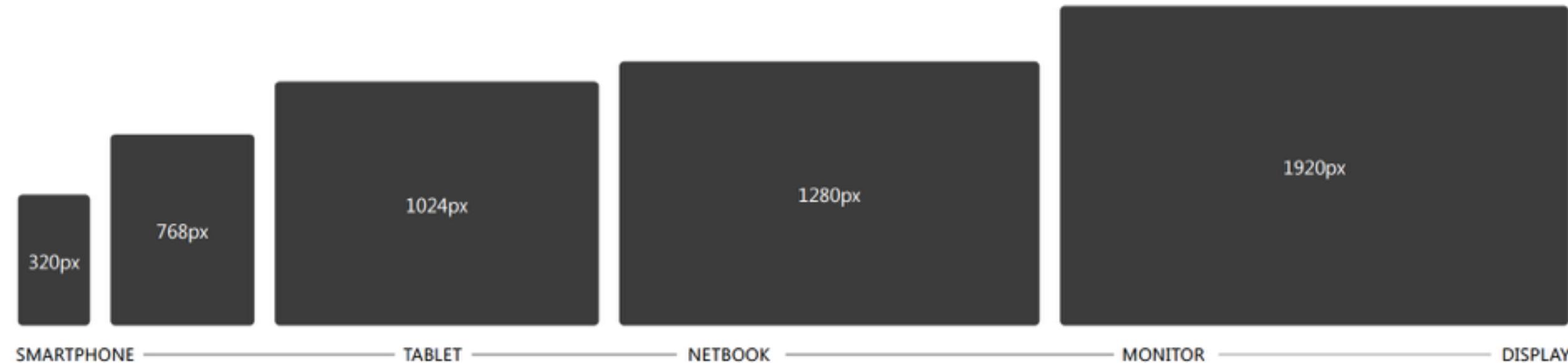
You put water into a cup it becomes **the cup**.  
You put water into a bottle it becomes **the bottle**.  
You put it in a teapot, it becomes **the teapot**.

josh clark [originally bruce lee] — seven deadly mobile myths

# 반응형 웹 시대를 위한 디자인

각기 다른 기기의 다른 비율과 해상도를 먼저 이해한 뒤, 스케치, 프로토타입, 시각화, 테스트 등을 진행해야 한다.  
우리가 할 일은 미래 친화적인 디자인 생태계를 만드는 것이다.

기술을 이용해, 동료를 이용해, 세상을 이용해

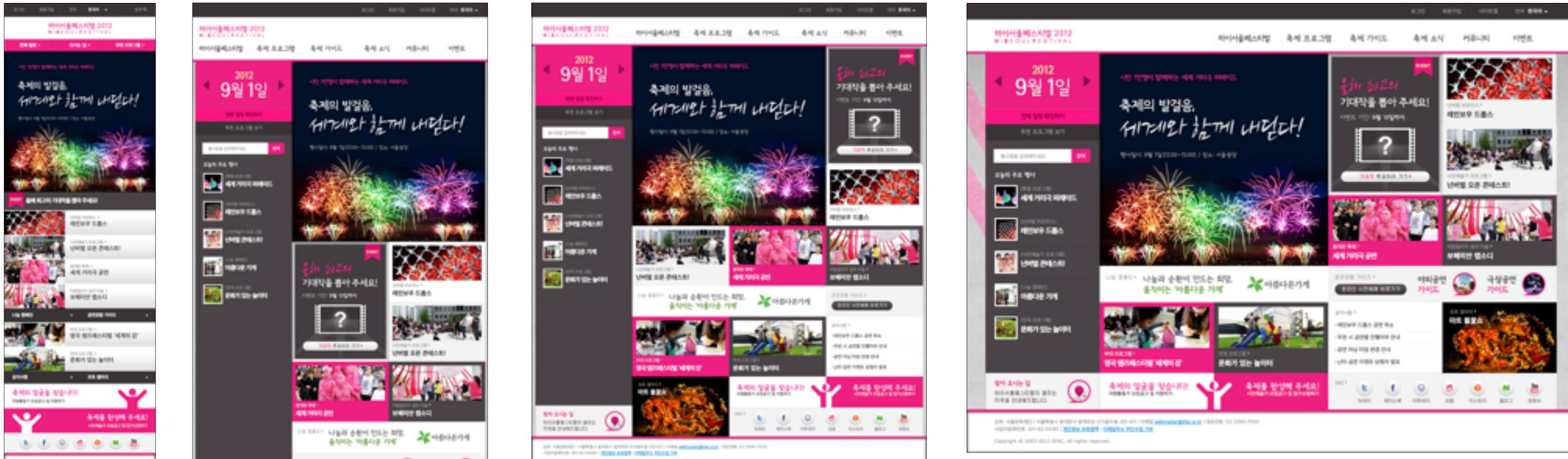


We need to **sketch, visualize, test, prototype** and understand the dynamic behaviors on different devices with different proportions & resolutions. Our job now is to **create future friendly design ecosystems**.

# 반응형 웹 디자인이란?

반응형 웹이 의미하는 바는 서비스를 디자인 하는 과정에  
사용자의 환경(스크린 사이즈, 플랫폼, 회전 방향 등)을 고려하여 응답할 수 있도록 제작하는 것을 말한다.

기술을 이름ake, 동료를 이름ake, 세상을 이름ake

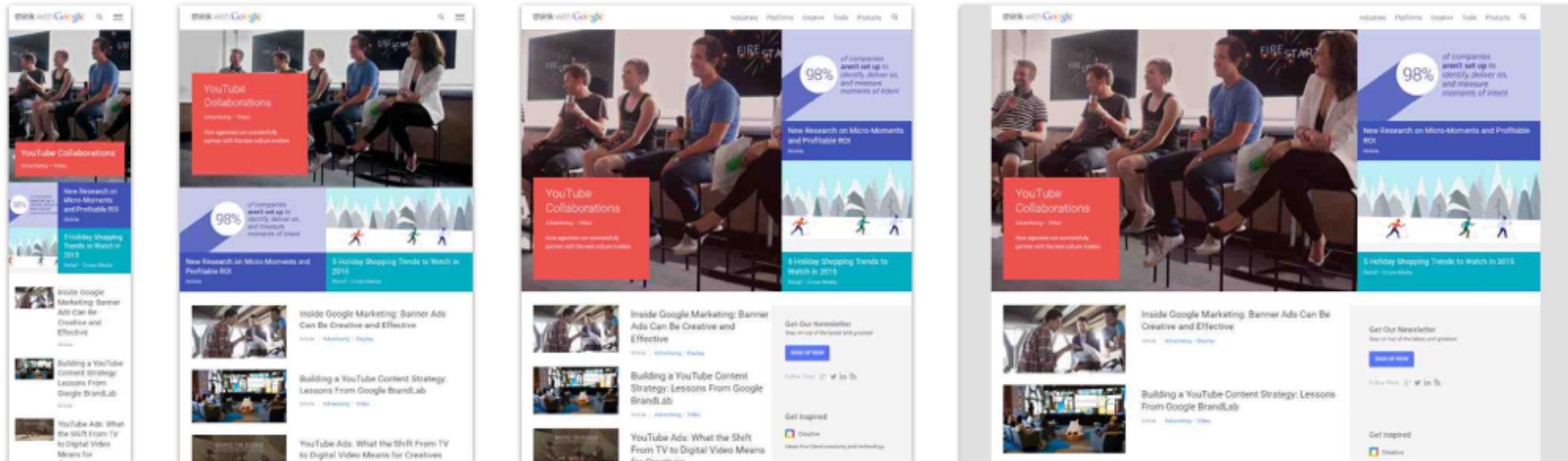


[iropke.com/proposal/HiSeoul](http://iropke.com/proposal/HiSeoul)

# 반응형 웹 디자인이란?

반응형 웹이 의미하는 바는 서비스를 디자인 하는 과정에  
사용자의 환경(스크린 사이즈, 플랫폼, 회전 방향 등)을 고려하여 응답할 수 있도록 제작하는 것을 말한다.

기술을 이름기, 동료를 이름기, 세상을 이름기



기술을 이용해, 동료를 이용해, 세상을 이용해



# 반응형 웹 시대의 서막

Ethan Marcotte  
[Responsive Web Design - A list apart](#)  
May 25, 2010



Rather than quarantining our content into disparate, device-specific experiences, we can use media queries to progressively enhance our work within different viewing contexts.

Ethan Marcotte



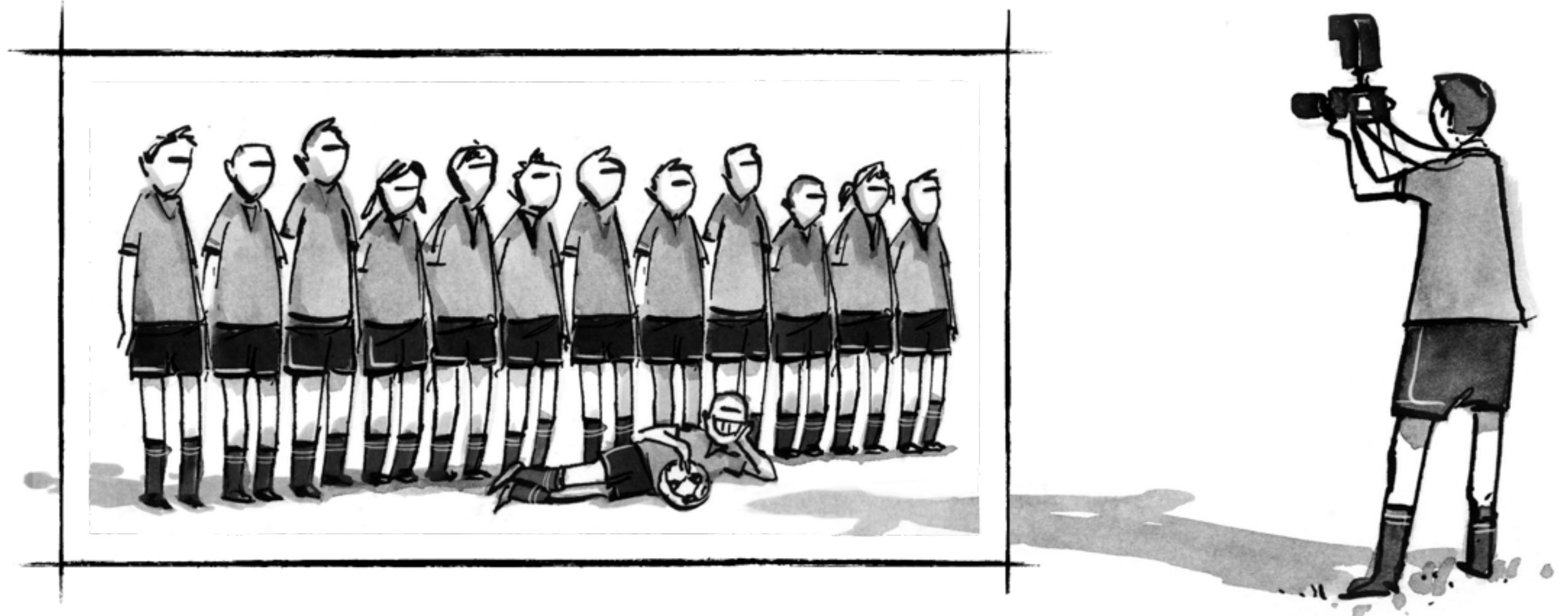
디바이스마다 콘텐츠를 감추거나 격리시키는 것보다는,  
각 기기의 경험을 맞춰 최적화된 뷰(View)를 사용자에게 제공하여  
콘텐츠를 효율적으로 담는 것이 우리가 수행해야 할 일입니다.

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

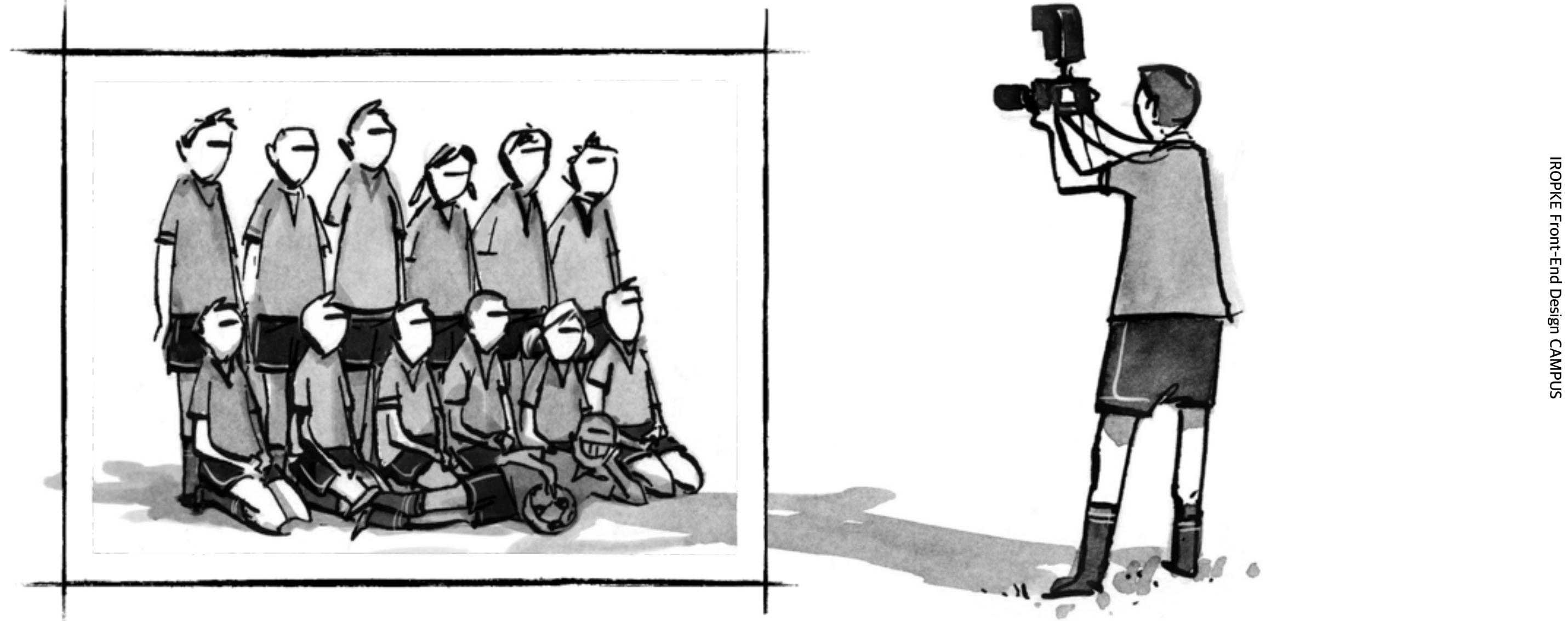


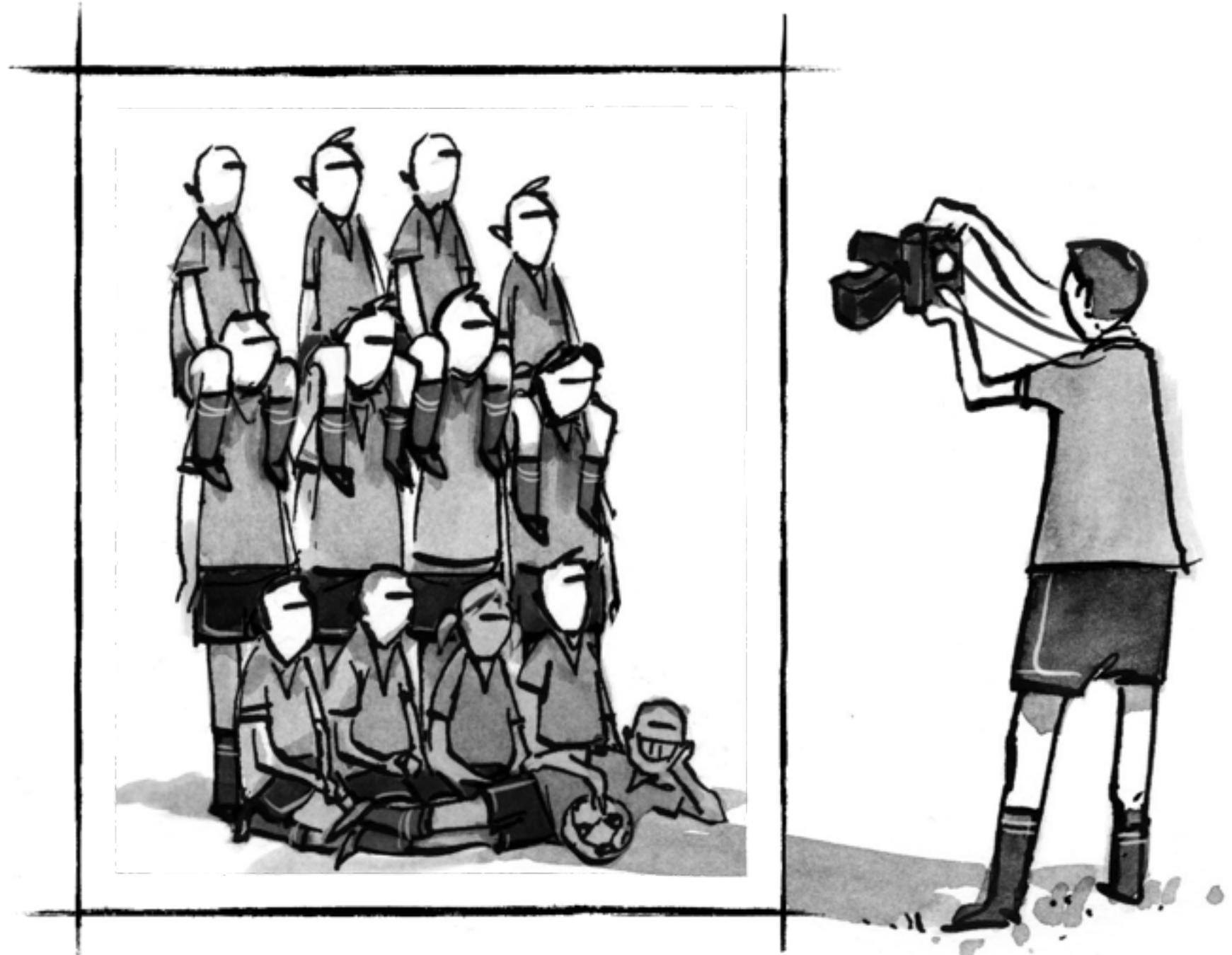
기술을 이용해, 동료를 이용해, 세상을 이용해



IROPKE Front-End Design CAMPUS

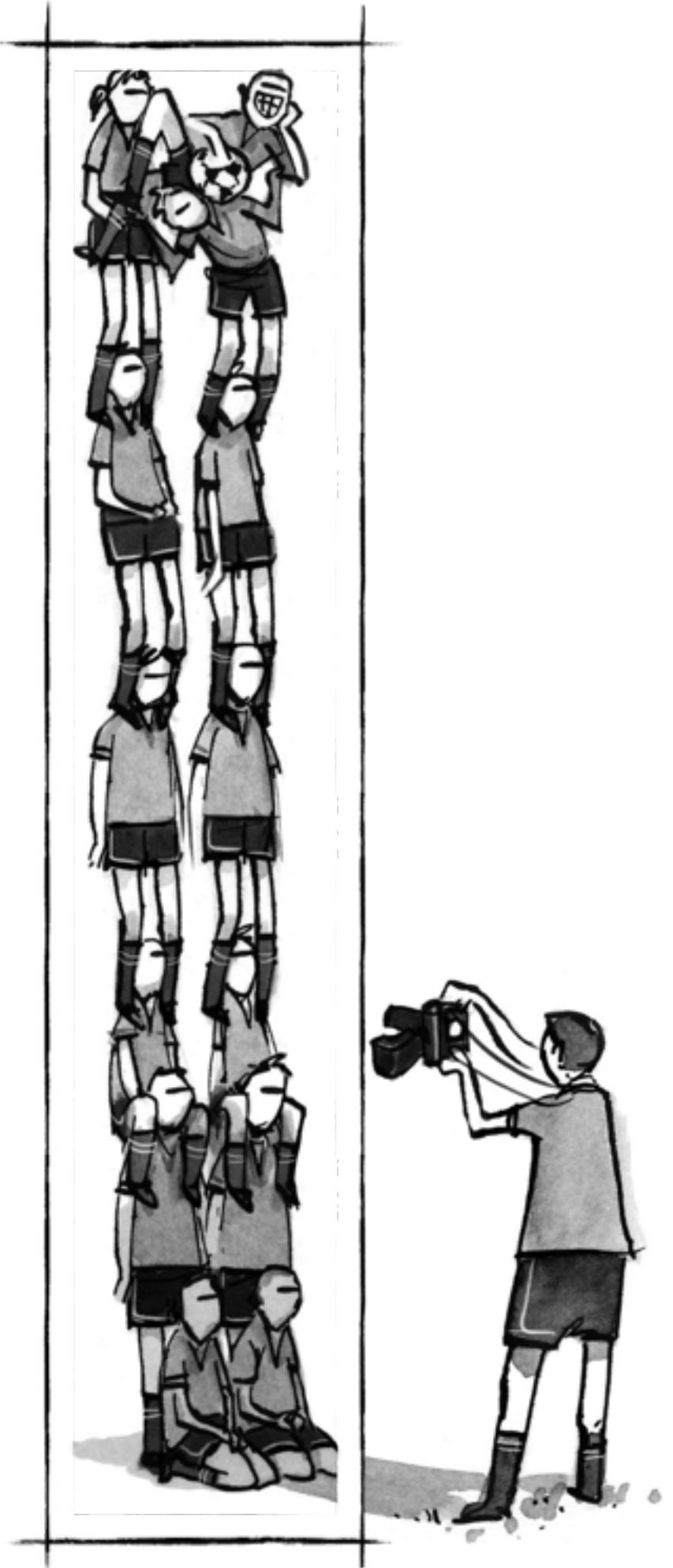
기술을 이용해, 동료를 이용해, 세상을 이용해





기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



# 반응형 웹 프로젝트를 시작하기 전 무엇을 고려하고, 알아야 할까?

# 반응형 웹 프로젝트를 시작하기 전 무엇을 고려하고, 알아야 할까?

당신의 커리어 전환점 패스트캠퍼스

콘텐츠 전략 (Content strategy)

유연한 그리드 레이아웃 (Flexible grid layout)

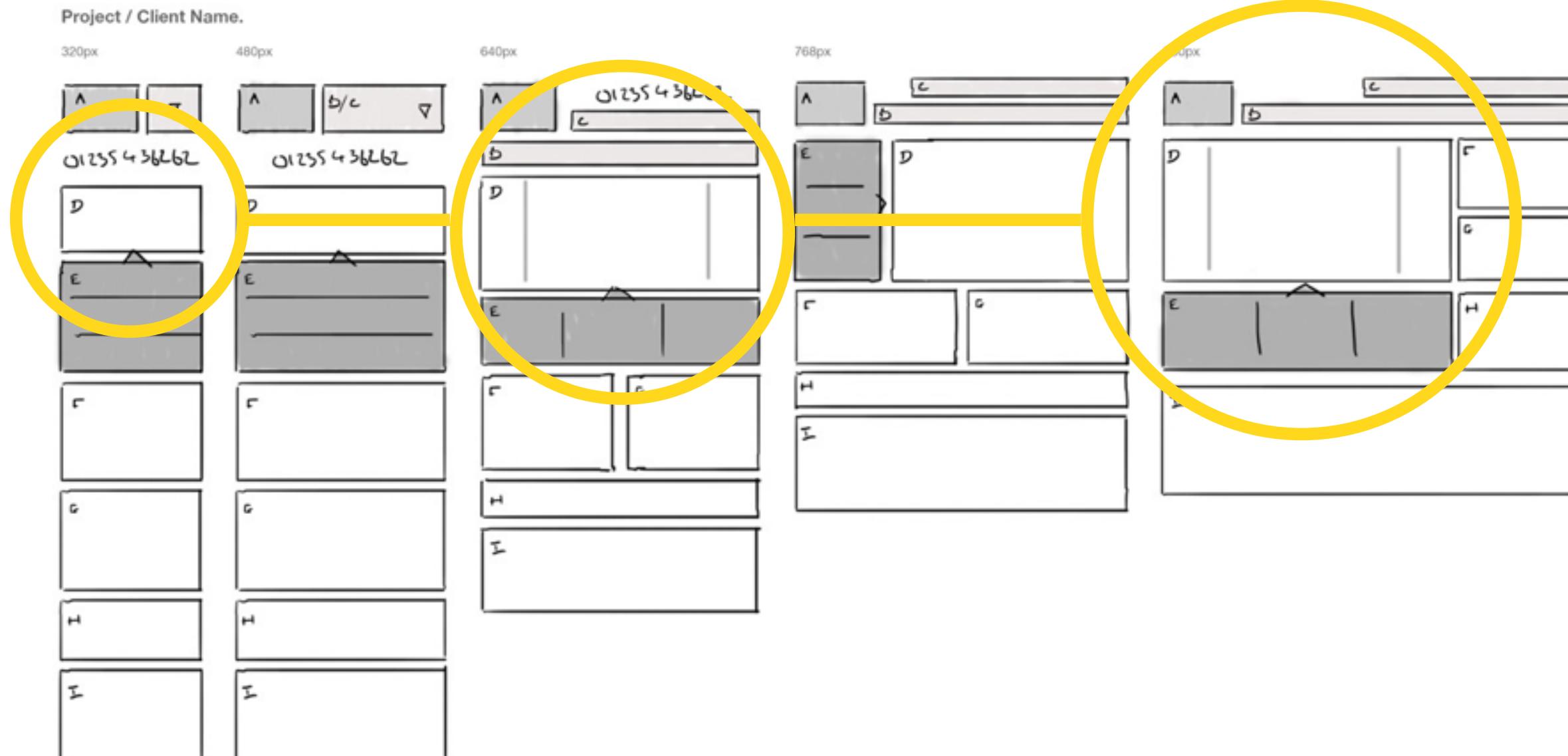
유연한 이미지 / 미디어 (Flexible images and media)

디바이스 픽셀 밀도 (Device Pixel Density)

중단점 / 미디어 쿼리 (Breakpoint and Media queries)

## 콘텐츠 전략

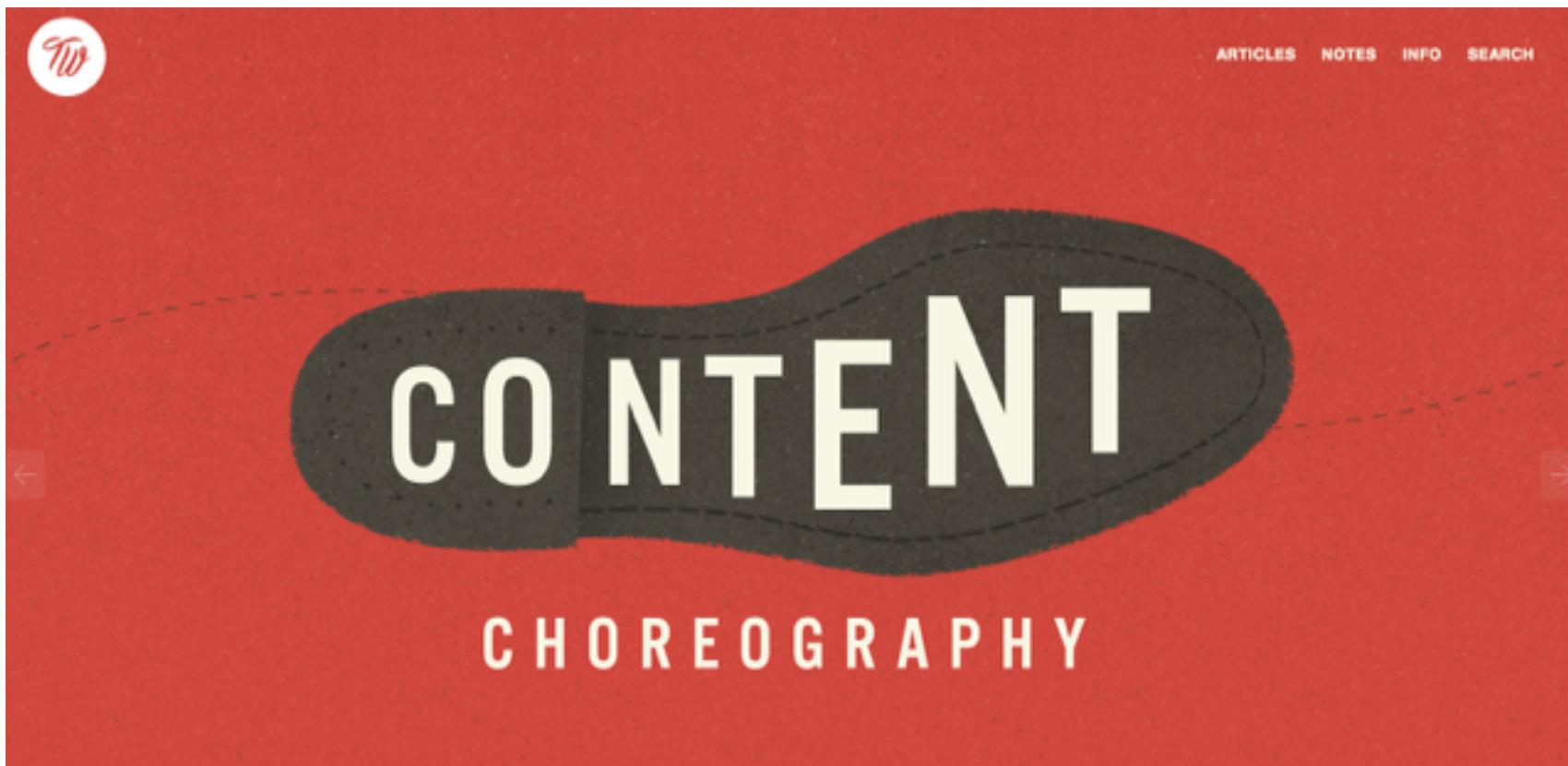
콘텐츠에 대한 사전 분석 없이 좋은 서비스는 제작할 수 없다. 각 뷰에서 그려질 콘텐츠 구조를 치밀하게 분석하지 않고는 좋은 사용자 경험을 제공하기 어렵다.





## 콘텐츠 구성

고정된 가로 폭과 해상도에서 콘텐츠가 영구적으로 자리 잡았던 디자인은 과거의 것이 되었다. 다양하게 변화되는 가로 폭과 해상도에 최적의 경험을 제공할 수 있는 무용술이 필요로 한다.



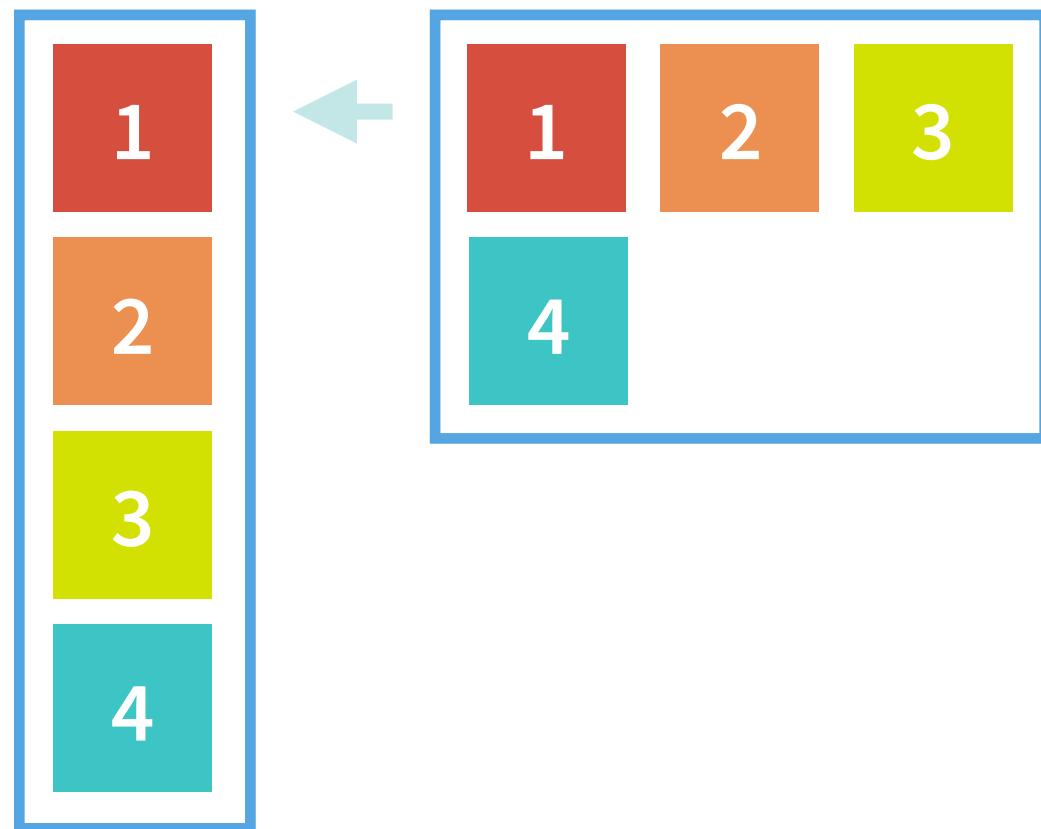
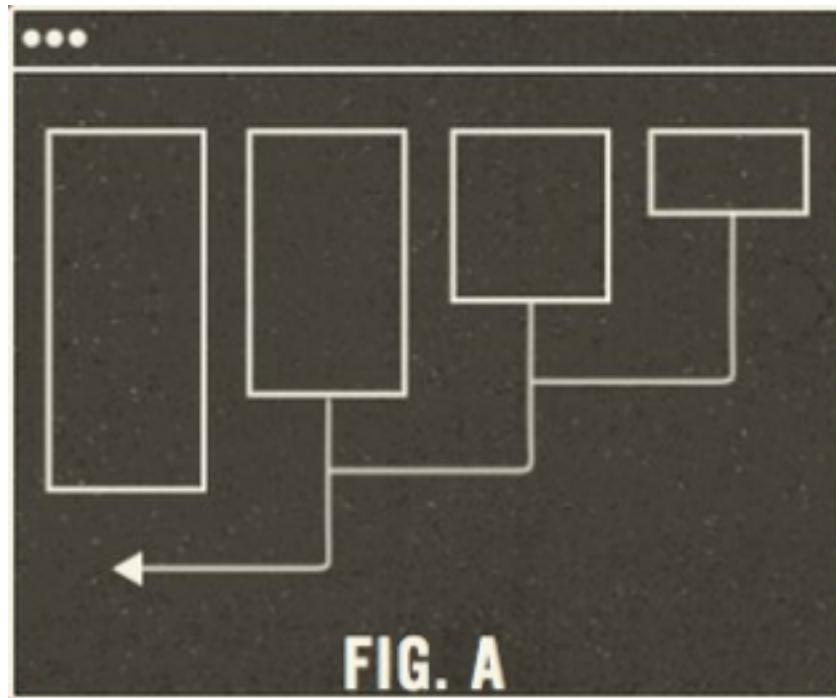
### choreography

1. 안무 2. 안무법 3. 무용술

미국 [kō(,:)riágrefi] ↗ 영국 [-óg-] ↗

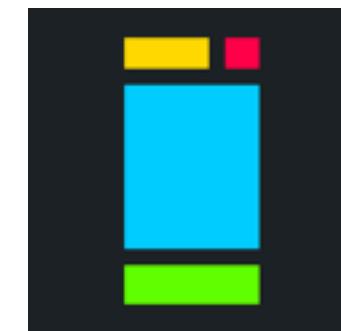
## 콘텐츠 쌓임

FIG. A 그림에서 보여지는 것처럼 4컬럼 레이아웃 디자인의 화면 폭이 줄어들 경우 쌓이는 것은 불가피 하다. ( $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$ ) 이와 같이 제작하는 것은 간단하지만, 화면 폭이 작은 환경에서 먼저 표시되는 1번이 가장 중요한 콘텐츠가 아닐 수도 있다.



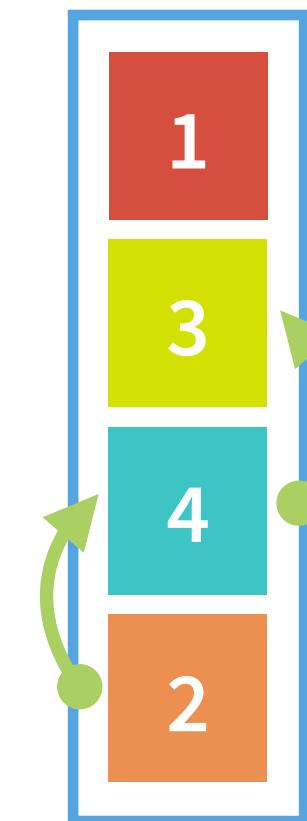
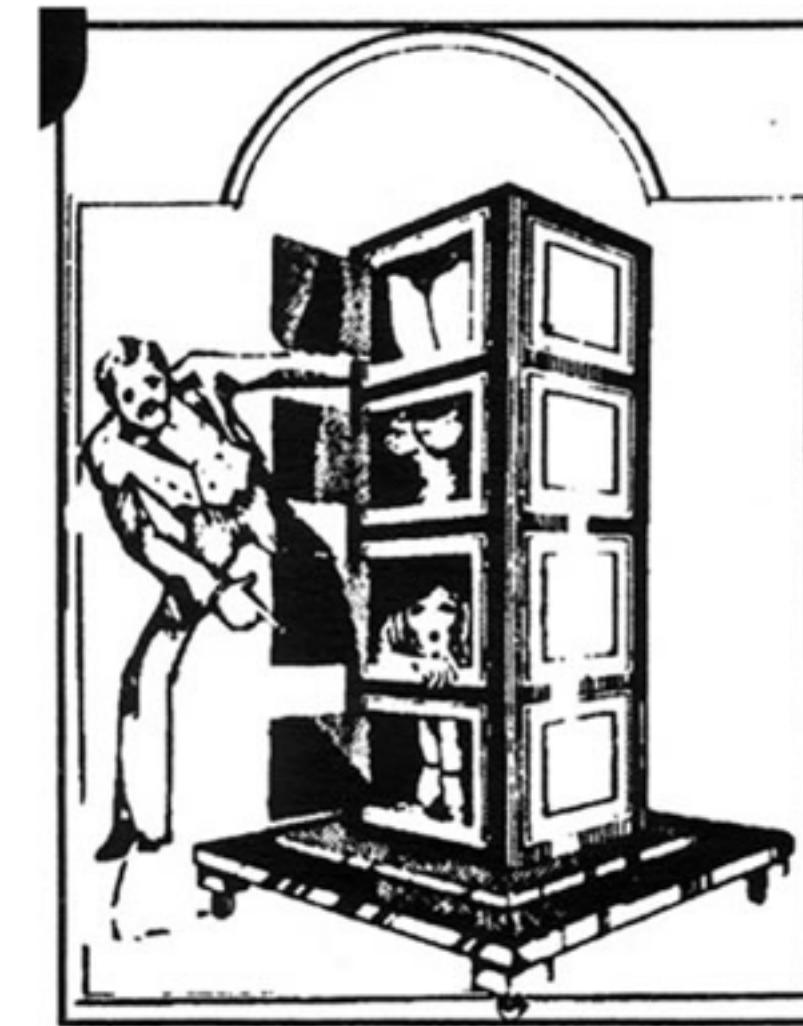
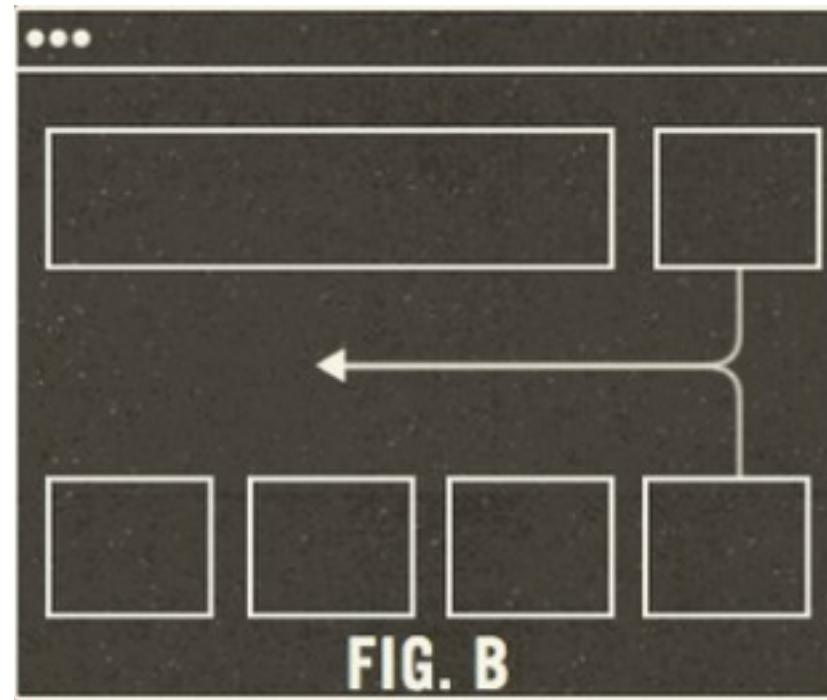
## 콘텐츠 순서

콘텐츠가 쌓이는 구조의 레이아웃이 아니라, 아래 예시 그림처럼 주요 콘텐츠가 보조 콘텐츠 보다 우선적으로 화면에 보여져야 한다. 이와 같은 접근은 화면 크기가 작은 모바일 환경에서 **우선적으로 보여(읽혀)져야 할 콘텐츠**가 무엇인지 고려해야 함을 말한다.



## 콘텐츠 맞물림

**FIG. B** 그림에서 보여지는 것처럼 폭이 좁고 길쭉한 뷰에서 각 콘텐츠 사이에 맞물리며 접하는 구조로 콘텐츠가 접힌다. 상황에 따라서는 순서가 뒤섞이는 구조로 보여질 경우도 있다.



## 플렉시블 박스

앞서 다룬 콘텐츠 중심 전략, 콘텐츠 구성(안무법) 등을 원활하게 구현하기 위한 새로운 레이아웃 기술이 요구되었다. 이에 탄생하게 된 레이아웃 모듈이 플렉시블 박스이다.

**W3C Candidate Recommendation**

**TABLE OF CONTENTS**

- 1 Introduction
- 1.1 Overview
- 1.2 Module interactions
- 2 Flex Layout Box Model and Terminology
- 3 Flex Containers: the 'flex' and 'inline-flex' 'display' values
- 4 Flex Items
  - 4.1 Absolutely-Positioned Flex Children
  - 4.2 Flex Item Margins and Paddings
  - 4.3 Flex Item Z-Ordering
  - 4.4 Collapsed Items
  - 4.5 Implied Minimum Size of Flex Items
- 5 Ordering and Orientation
  - 5.1 Flex Flow Direction: the 'flex-direction' property
  - 5.2 Flex Line Wrapping: the 'flex-wrap' property
  - 5.3 Flex Direction and Wrap: the 'flex-flow' shorthand
  - 5.4 Display Order: the 'order' property
    - 5.4.1 Reordering and Accessibility
- 6 Flex Lines

## CSS Flexible Box Layout

### Module Level 1

W3C Candidate Recommendation, 26 May 2016

This version: <http://www.w3.org/TR/2016/CR-css-flexbox-1-20160526/>

Latest published version: <http://www.w3.org/TR/css-flexbox-1/>

Editor's Draft: <https://drafts.csswg.org/css-flexbox/>

Previous Versions:

<http://www.w3.org/TR/2016/CR-css-flexbox-1-20160301/>  
<http://www.w3.org/TR/2015/WD-css-flexbox-1-20150514/>  
<http://www.w3.org/TR/2014/WD-css-flexbox-1-20140925/>  
<http://www.w3.org/TR/2014/WD-css-flexbox-1-20140325/>  
<http://www.w3.org/TR/2012/CR-css3-flexbox-20120918/>  
<http://www.w3.org/TR/2012/WD-css3-flexbox-20120612/>  
<http://www.w3.org/TR/2012/WD-css3-flexbox-20120322/>  
<http://www.w3.org/TR/2011/WD-css3-flexbox-20111129/>  
<http://www.w3.org/TR/2011/WD-css3-flexbox-20110322/>  
<http://www.w3.org/TR/2009/WD-css3-flexbox-20090723/>

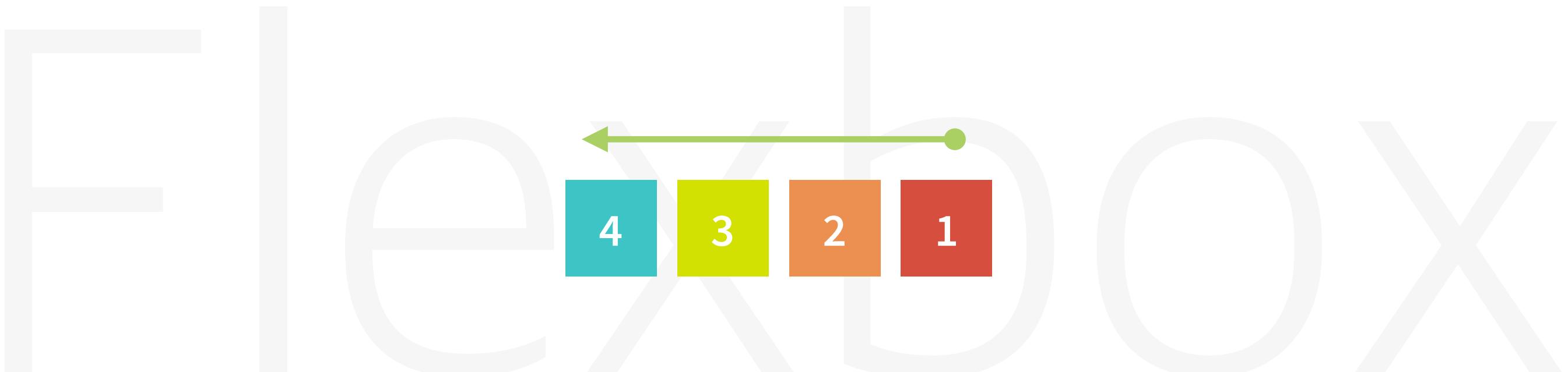
Feedback: [www-style@w3.org](mailto:www-style@w3.org) with subject line "[css-flexbox] ... message topic ..." ([archives](#))

Test Suite: [http://test.csswg.org/suites/css-flexbox-1\\_dev/nightly-unstable/](http://test.csswg.org/suites/css-flexbox-1_dev/nightly-unstable/)

Fast campus

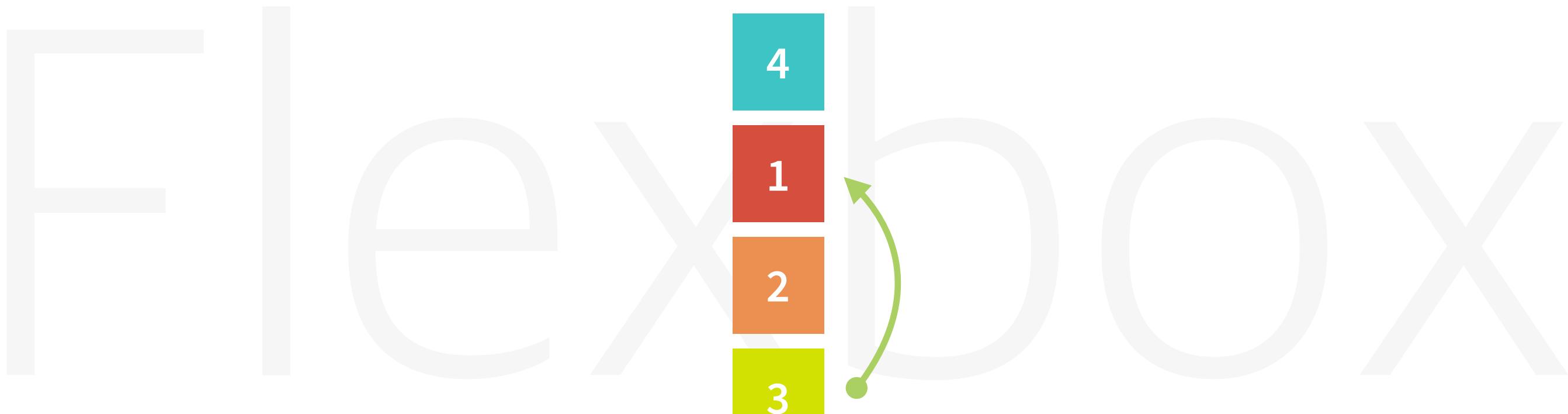
FAST CAMPUS SCHOOL 2016  
Copyright FAST CAMPUS Corp. All Rights Reserved





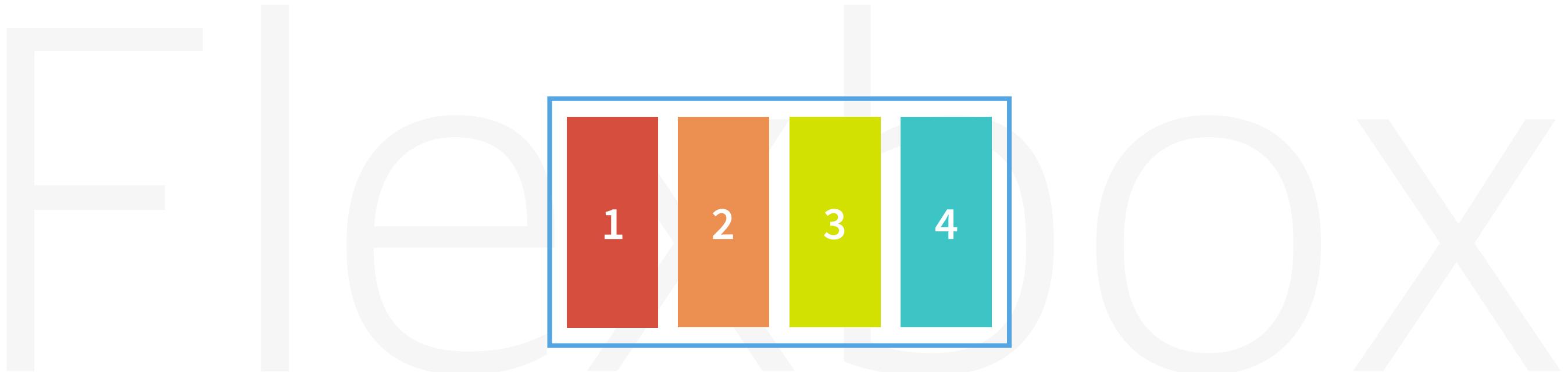


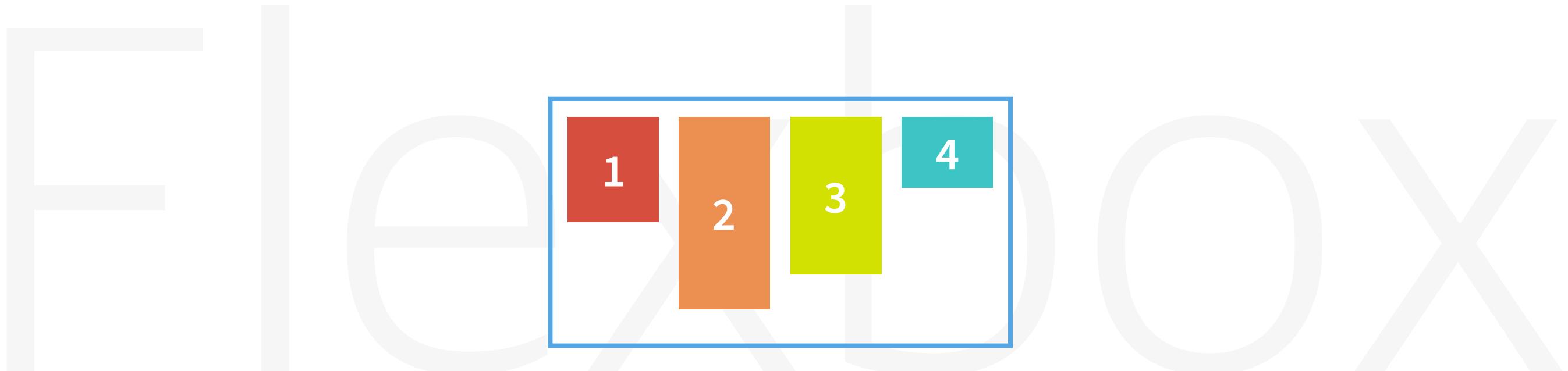








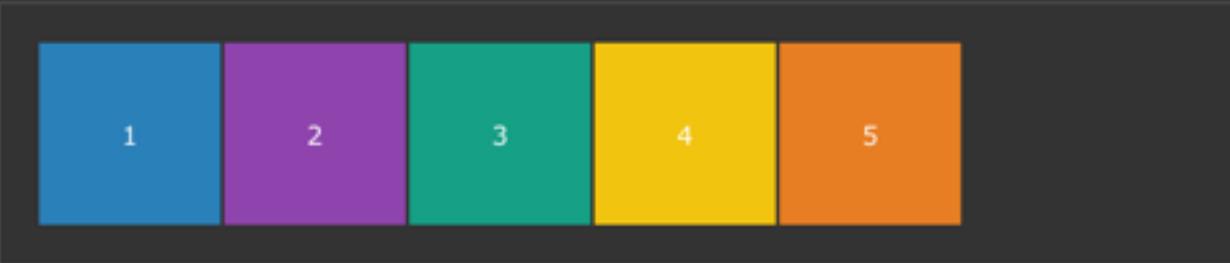




## Properties for the flex container

**FLEX-DIRECTION** ( property of the flex container )

row:  row-reverse:  column:  column-reverse:



**FLEX-WRAP** ( property of the flex container )

nowrap:  wrap:  wrap-reverse:



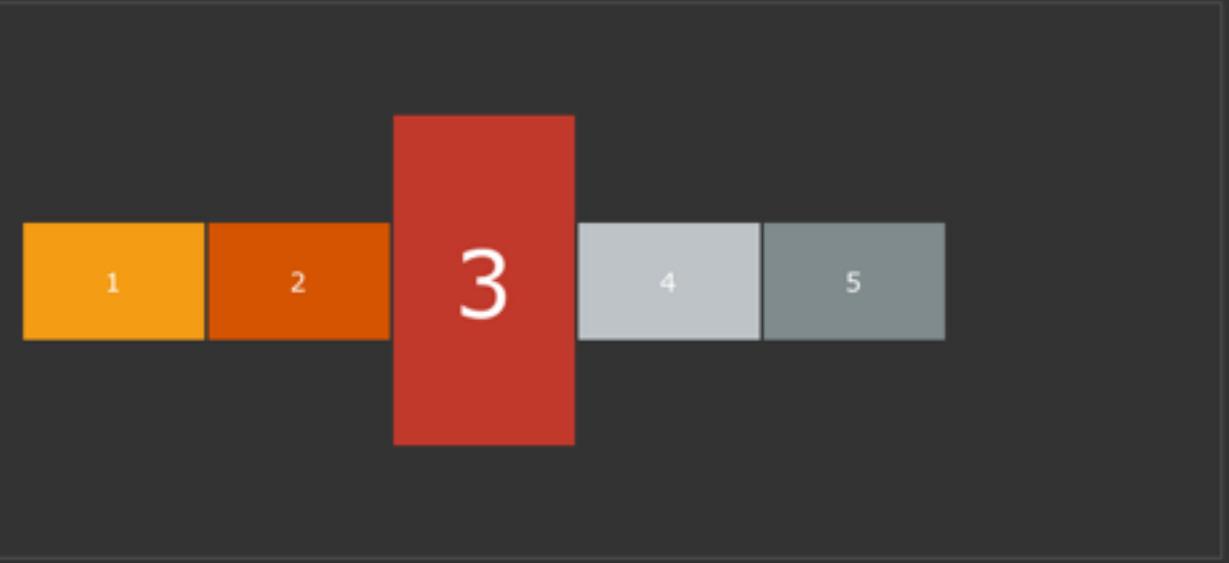
**ALIGN-ITEMS** ( property of the flex container )

flex-start:  flex-end:  center:  baseline:  stretch:

## Properties for the flex items

**ALIGN-SELF** ( property for flex items )

auto:  flex-start:  flex-end:  center:  baseline:  stretch:



**FLEX-GROW** ( property for flex items )

elemento 1:  elemento 2:  elemento 3:  elemento 4:   
elemento 5:



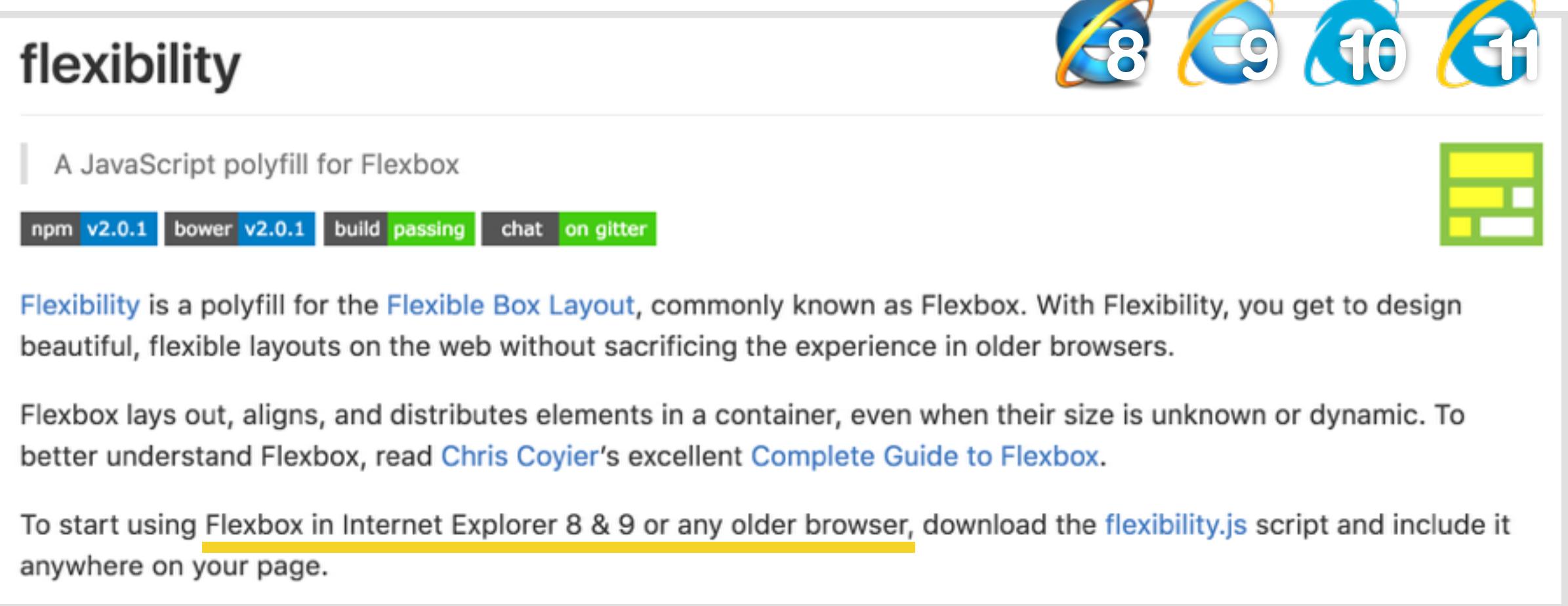
## 플렉시블 박스 호환성

2016년 현재 플렉시블 박스 호환성을 살펴보면 바로 사용 가능하다. 단, 데스크탑 환경 IE 경우는 버그가 존재하기에 플렉시블 박스 레이아웃 사용에 주의가 요구된다.



## 플렉시빌리티 대체수단

문제가 되는 데스크탑 IE 환경의 경우는 자바스크립트를 활용한 대체 라이브러리를 사용하여 문제를 해결할 수 있다. (스크립트 사용에 따른 성능 문제 검토 요구됨)



**flexibility**

A JavaScript polyfill for Flexbox

npm v2.0.1 bower v2.0.1 build passing chat on gitter

Flexibility is a polyfill for the [Flexible Box Layout](#), commonly known as Flexbox. With Flexibility, you get to design beautiful, flexible layouts on the web without sacrificing the experience in older browsers.

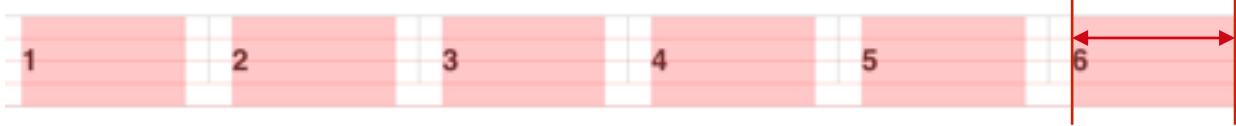
Flexbox lays out, aligns, and distributes elements in a container, even when their size is unknown or dynamic. To better understand Flexbox, read [Chris Coyier's excellent Complete Guide to Flexbox](#).

To start using Flexbox in Internet Explorer 8 & 9 or any older browser, download the [flexibility.js](#) script and include it anywhere on your page.

## 유연한 그리드 레이아웃

유연한 그리드(Fluid Grid)는 일관되게 콘텐츠를 구성하지만, 디스플레이 사이즈에 따라 크기와 위치가 변경된다. 유연한 그리드를 구현하는 방법은 고정 그리드를 기반으로 상대적인 수치를 도출하는 것에서 시작된다.

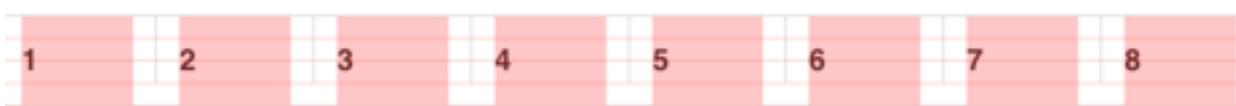
6-Column Grid



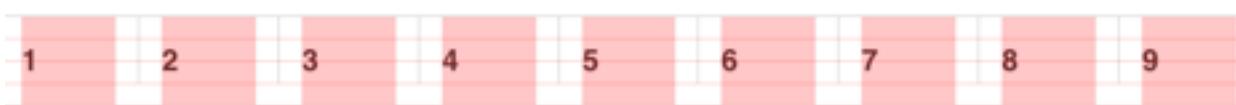
7-Column Grid



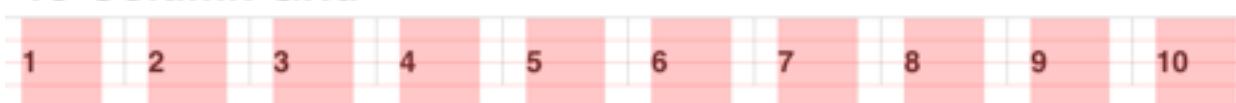
8-Column Grid



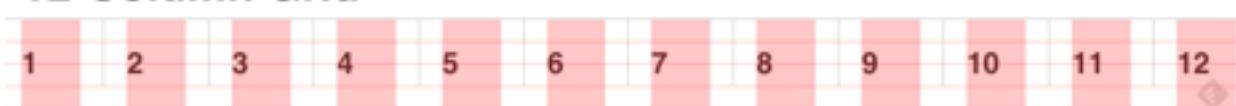
9-Column Grid



10-Column Grid



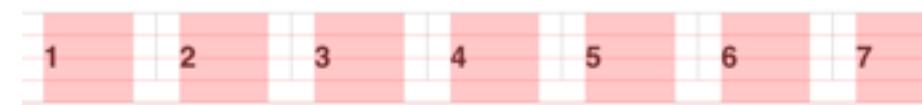
12-Column Grid



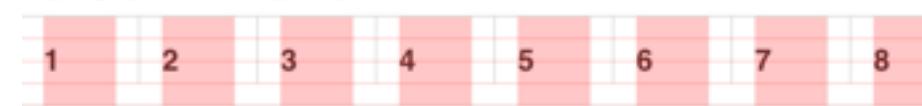
6-Column Grid



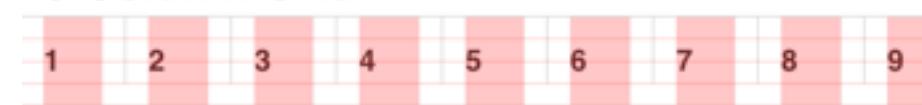
7-Column Grid



8-Column Grid



9-Column Grid



10-Column Grid



12-Column Grid



## 유연한 그리드 레이아웃

RWD에서는 픽셀(Pixel)이 아닌, 상대 단위(em, rem, % 등)를 사용해야 하기에  
픽셀을 상대 단위로 바꾸는 계산식을 사용해야 한다.

### Target : context = results

$$\frac{24 : 16 = 1.5\text{em}}{700 : 988 = 0.7085}$$

Fontsize

DIV

```
h1 {  
    font-family: Georgia, serif;  
    font-size: 1.5em;      /* 24px / 16px = 1.5em */  
}
```

```
h1 {  
    width: 70.85%;  
    /* 700px / 988px = 0.7085 */
```



## 유연한 그리드 레이아웃

당신의 커리어 전환점 페스트 캠퍼스



Illustration by Kevin Cornell

Issue № 279

### Fluid Grids

by Ethan Marcotte · March 03, 2009

# THE RATIO REVOLUTION WILL NOT BE TELEVISED

A BLOG ENTRY.

## Anyone else tired of Helvetica?

I mean it: I've got your *Neue Haas Grotesk* right here, buddy. Along with some lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Yeah, you heard me.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Posted on 15 February 2009. 15 comments so far.

Tagged with [rants](#), [typography](#), [filler copy](#), [sunshine](#) and [puppies](#).



# 유연한 그리드 레이아웃

당신의 커리어 전환점 페스트 캠퍼스

A BLOG ENTRY.

## THE RATIO REVOLUTION WILL NOT BE TELEVISED

### Anyone else tired of Helvetica?

I mean it: I've got your *Neue Haas Grotesk* right here, buddy. Along with some lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Yeah, you heard me.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

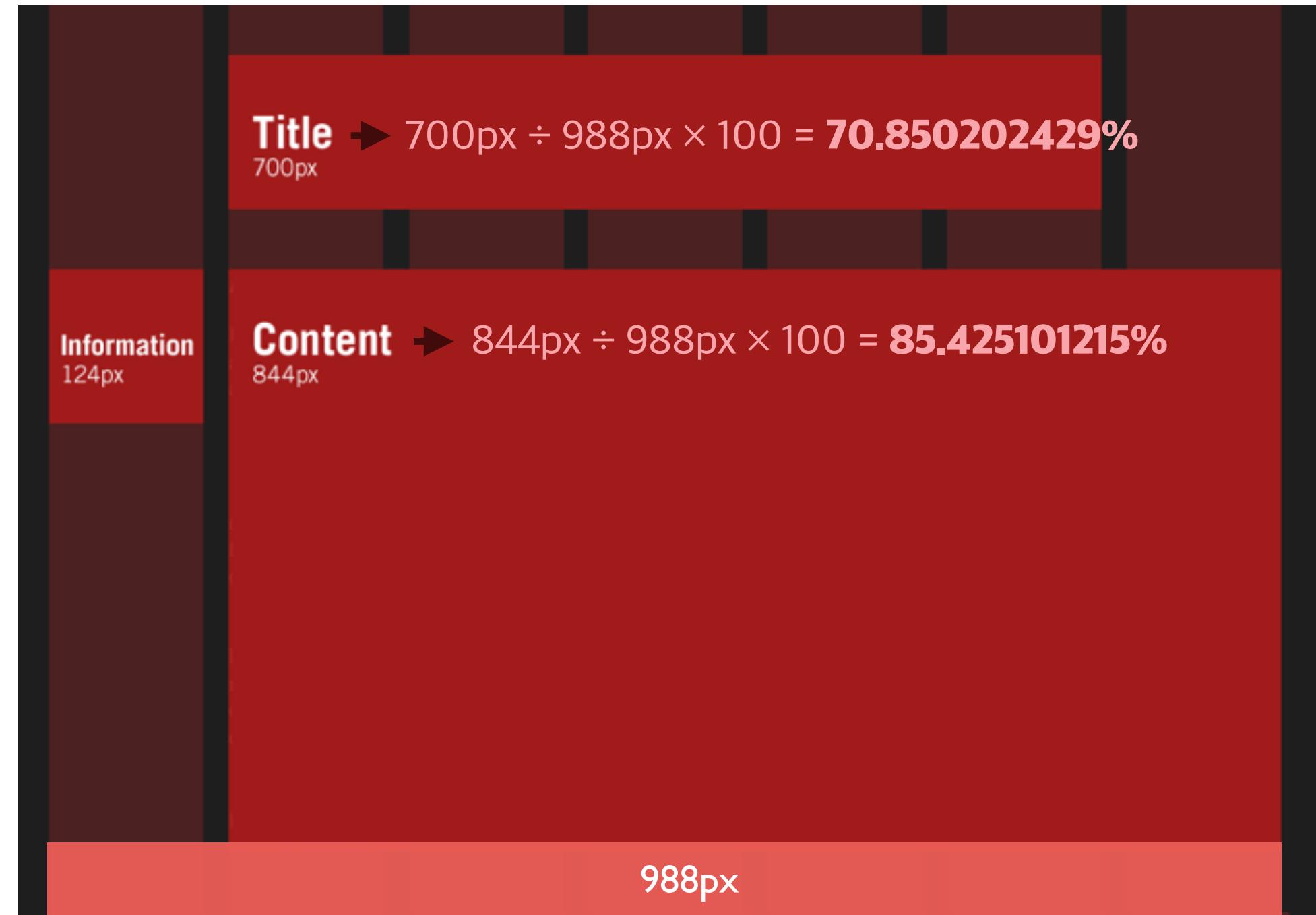
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

  Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Posted on 15 February 2009. 15 comments so far.

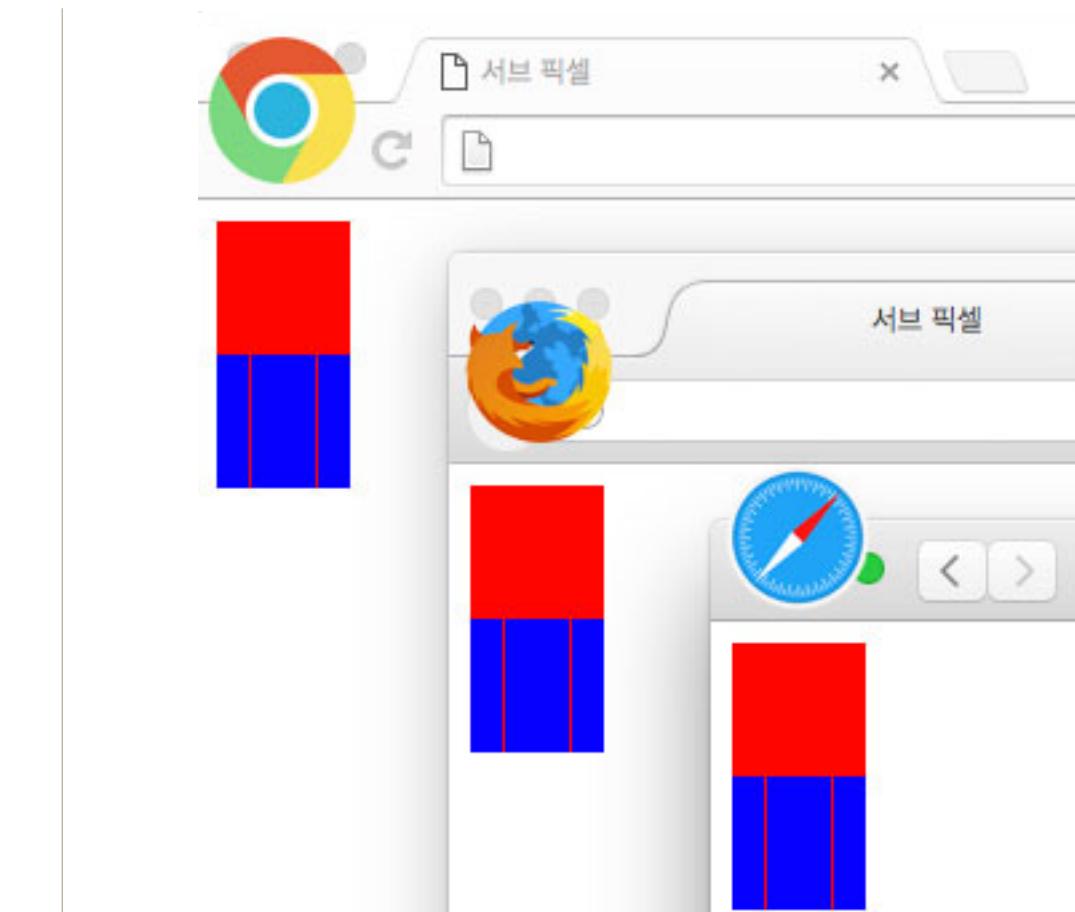
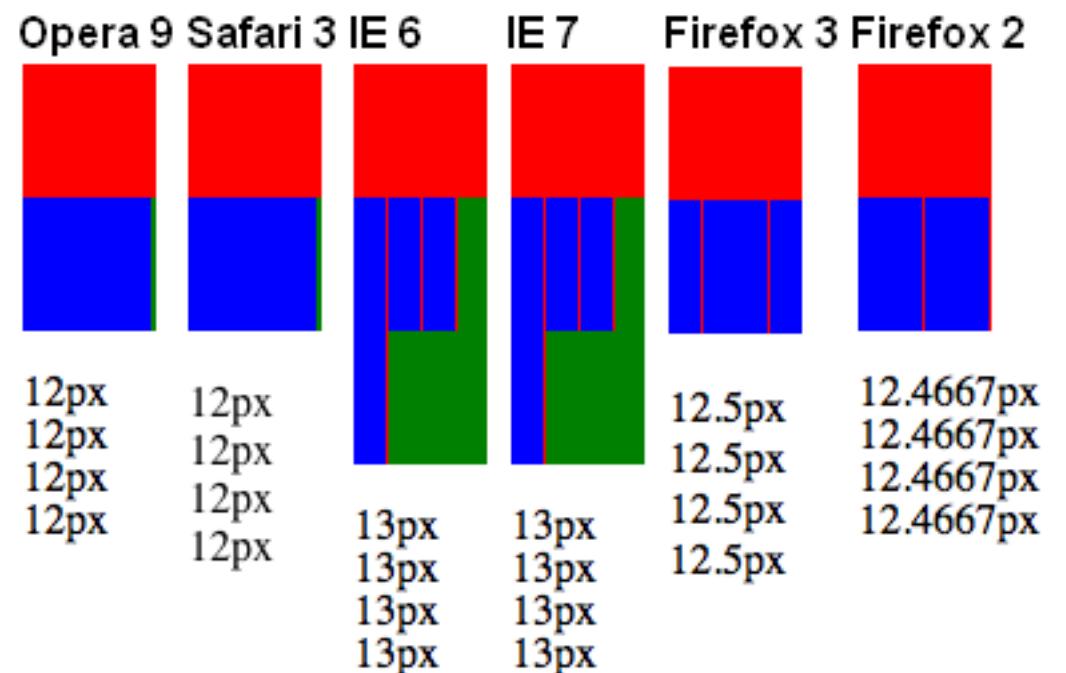
Tagged with [rants](#), [typography](#), [filler copy](#), [sunshine](#) and [puppies](#).

# 유연한 그리드 레이아웃

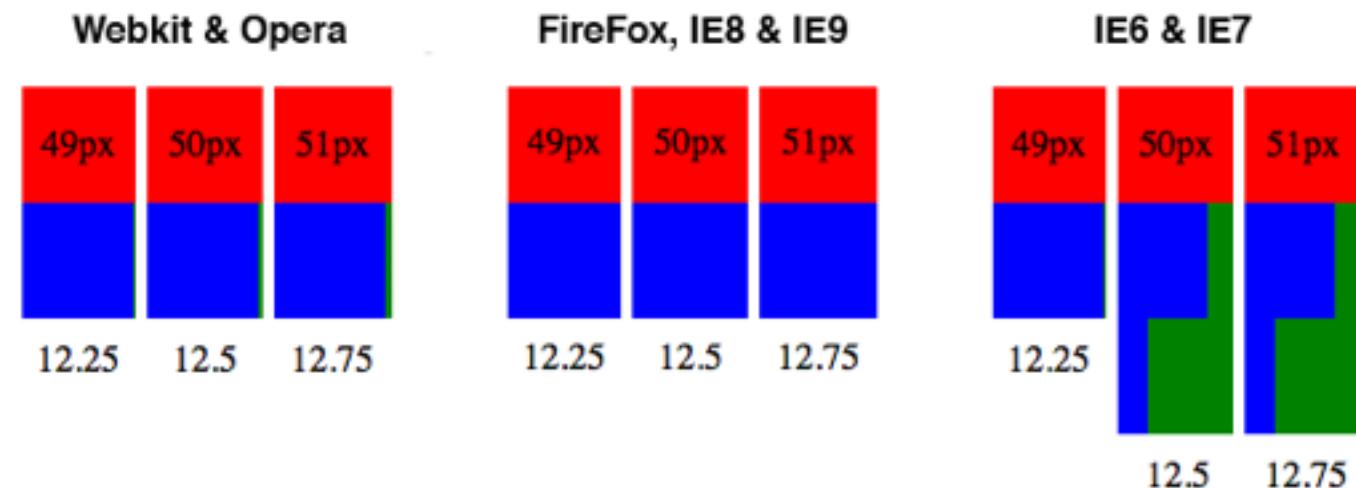


## 테크니컬 이슈

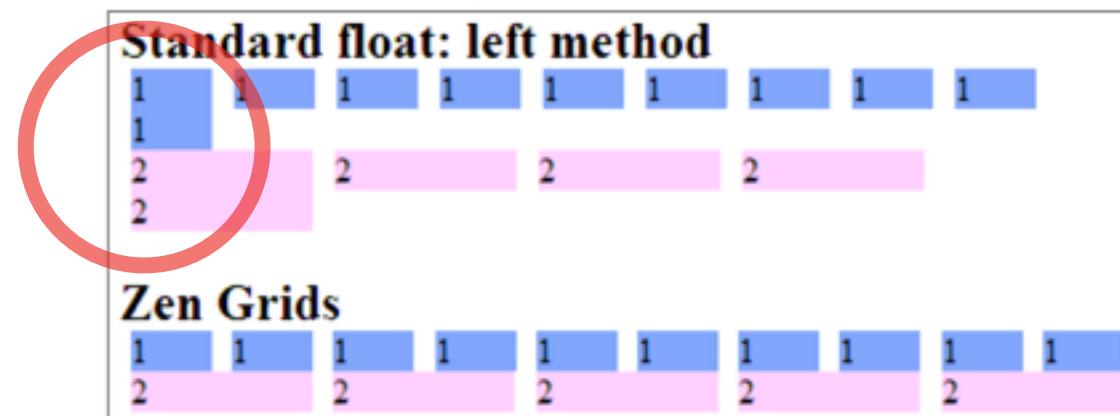
물처럼 흐르는 유연한 레이아웃(Fluid Layout)을 구현할 경우 발생하는 테크니컬 이슈는 웹 브라우저가 퍼센트(%) 값을 픽셀(px) 값으로 변경하는 과정에서 발생한다. 정확하게 정수로 떨어지지 않는 픽셀의 경우 각 브라우저마다 처리하는 방식이 다르기 때문.



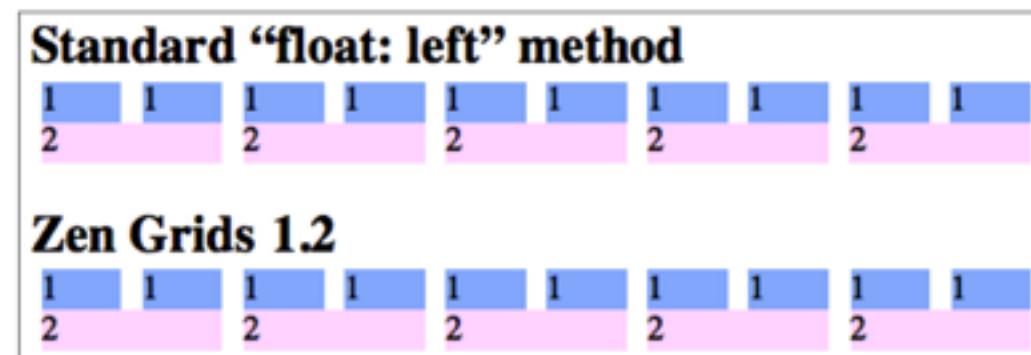
## 테크니컬 이슈



Screenshot of above page in IE 7



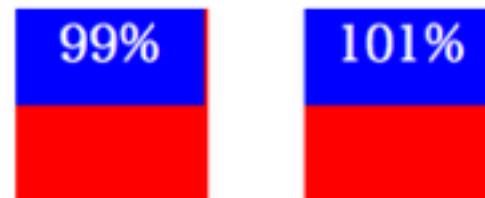
Screenshot of above page in Firefox 13. Note the sub-pixel rendering



# 테크니컬 이슈

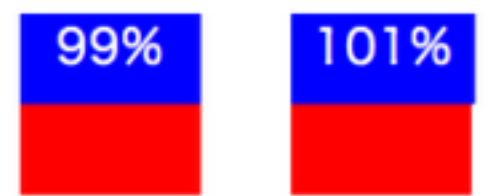
## Safari (OS X 및 iOS) Android 브라우저의 거동

먼저 Webkit 계의 거동에 대해 설명합니다. 먼저도 올렸다대로 Webkit 기반의 브라우저는 소수점 이하를 버리고 표시하기 위해 49.5px은 49px로 50.5px은 50px되고, 미만인 경우에는 약간 틈새가 생겨 나온 경우는 그냥 맞게 렌더링 됩니다.



## Chrome (mobile), Opera, IE의 거동

그러면, 앞의 예에서는 능숙한 상태로 조정 해 주었다 Chrome 등의 브라우저의 행동은 어떻게 될까요. 결론부터 말하면 일반적 Firefox 이외는 49.5px 이상 50.5 미만이면 50px로 렌더링됩니다. 하면 데모 페이지의 예에서 말하면 49.5px의 경우는 잘 들어가, 50.5px의 경우 1px 벗어나 표시됩니다. 덧붙이면, 49.5 미만의 경우는 1px 틈새가 있는 것처럼 보입니다. 아래는 Opera에서 렌더링 예입니다.



또한 위의 브라우저에서도 각각 인식 할 수있는 자리수가 다르기 때문에 예를 들어 `width : 100.998877 %` 등하면 세세하게 달라집니다. 인식 할 수있는 자릿수 내용은 CSS의 소수점 이하의 수치를 각 브라우저는 어떻게 해석할지 (<http://uninformedbuilding.com/articles/after-the-decimal-point-in-css/>)에서 자세히 설명되어 있습니다.

## 테크니컬 이슈

웹 환경 이슈(서브 픽셀 렌더링/CSS 개체 모델)을 이해한다면 레이아웃을 디자인할 때 다양한 브라우저와 장치에서 사이트가 제대로 표시되도록 몇 픽셀의 여유 공간을 두는 것이 좋다.



일반적으로 웹에서 픽셀 퍼펙트 레이아웃은 콘텐츠의 접근성, 호환성, 적응성이라는 목표와 상충되기 쉽기 때문에 결과론적으로 보면 최상의 선택이 아닙니다. 위 이미지들은 웹 디자이너가 픽셀 퍼펙트 디자인을 만들고자 했으나 예상치 못한 웹 플랫폼의 차이로 인해 버그가 발생하여 디자인이 엉망이 된 예입니다.

### 서브 픽셀 렌더링 및 CSS 개체 모델

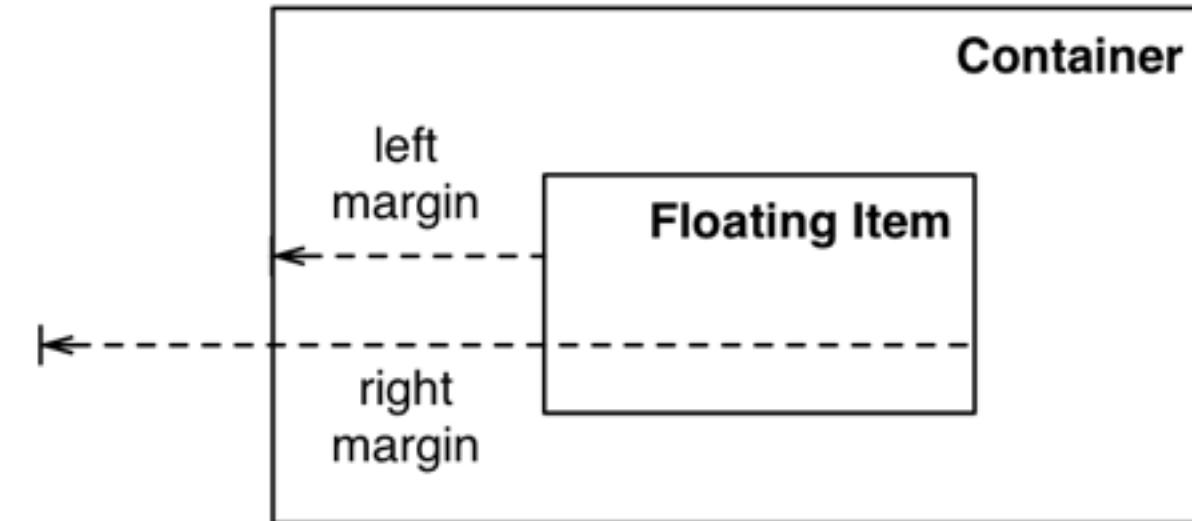
[ieblog](#) February 27, 2012

## 테크니컬 솔루션

플로트(float) 속성과 퍼센트(%) 속성 값으로 대변되는 RWD에서는 앞서 다룬 기술적 문제가 발생하게 되었고, 문제를 해결하기 위해 고안된 기술이 격리된 플로트(Isolate Float) 방법이다.

## Responsive Design's Dirty Little Secret

by John Albin Wilkins



## 테크니컬 솔루션

플로트(float) 속성과 퍼센트(%) 속성 값으로 대변되는 RWD에서는 앞서 다룬 기술적 문제가 발생하게 되었고, 문제를 해결하기 위해 고안된 기술이 격리된 플로트(Isolate Float) 방법이다.

### Isolate Float 테크닉

#### 테크닉 원리

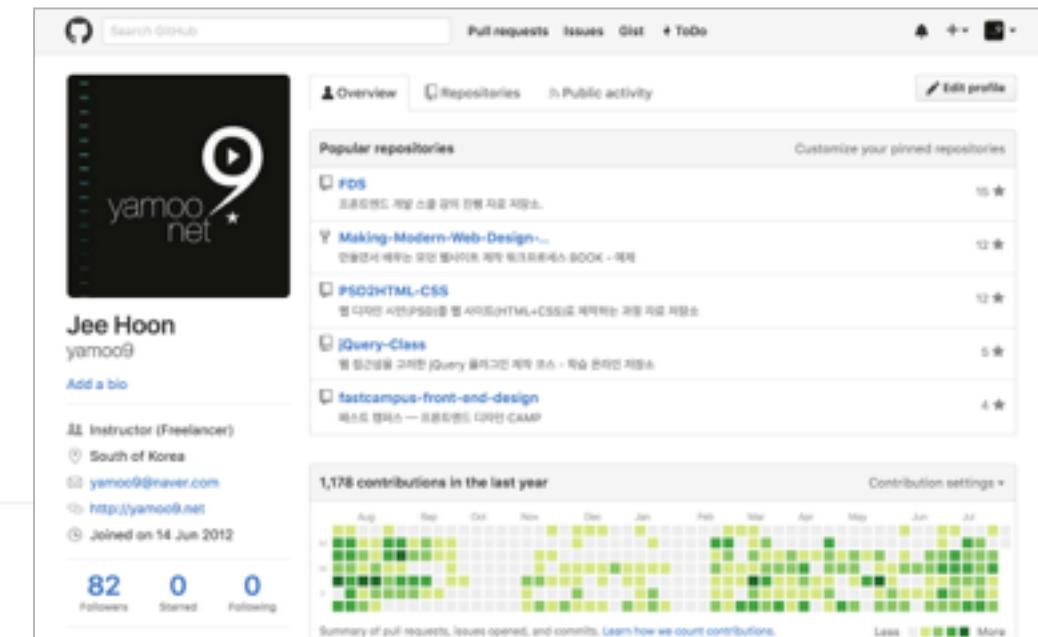
##### isolate 테크닉에 요구되는 사항

\* 테크닉 사용 시 혼란이 올 수 있으니, 음수 마진 값 설정에 대한 이해가 요구됨.

- 개별 레이아웃 요소는 `margin-left` 값을 통해 정확한 위치를 설정.
- 개별 레이아웃 요소는 `margin-right: -100%`; 를 사용하여 뒤에 오는 요소를 끌어(pull) 당긴다.

#### STEP 1

3 컬럼 레이아웃을 디자인한다고 가정.



## 유연한 이미지

유연한 이미지(Fluid Image)는 이미지를 포함하는 컨테이너 요소의 폭에 맞춰 크기가 변경되는 이미지를 말한다.



Issue № 328

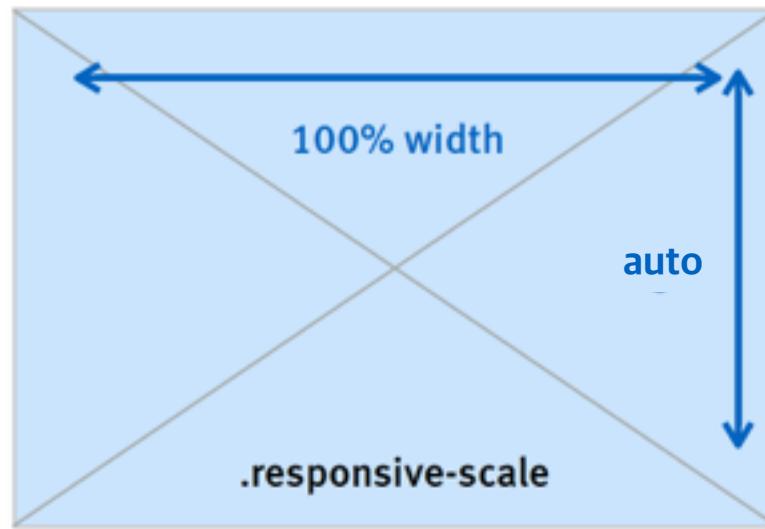
### Fluid Images

by Ethan Marcotte · June 07, 2011

## 유연한 이미지

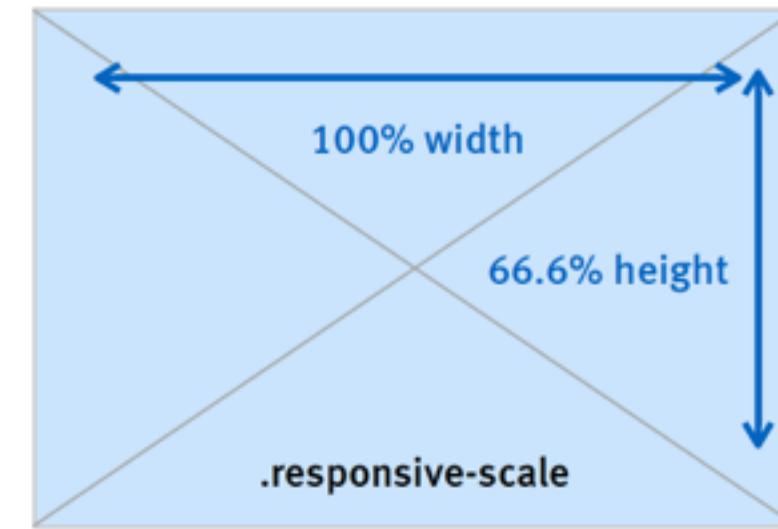
유연한 이미지(Fluid Image)는 이미지를 포함하는 컨테이너 요소의 폭에 맞춰 크기가 변경되는 이미지를 말한다.

콘텐츠 이미지



```
.responsive-scale {  
    width: 100%;  
    height: auto;  
}
```

배경 이미지

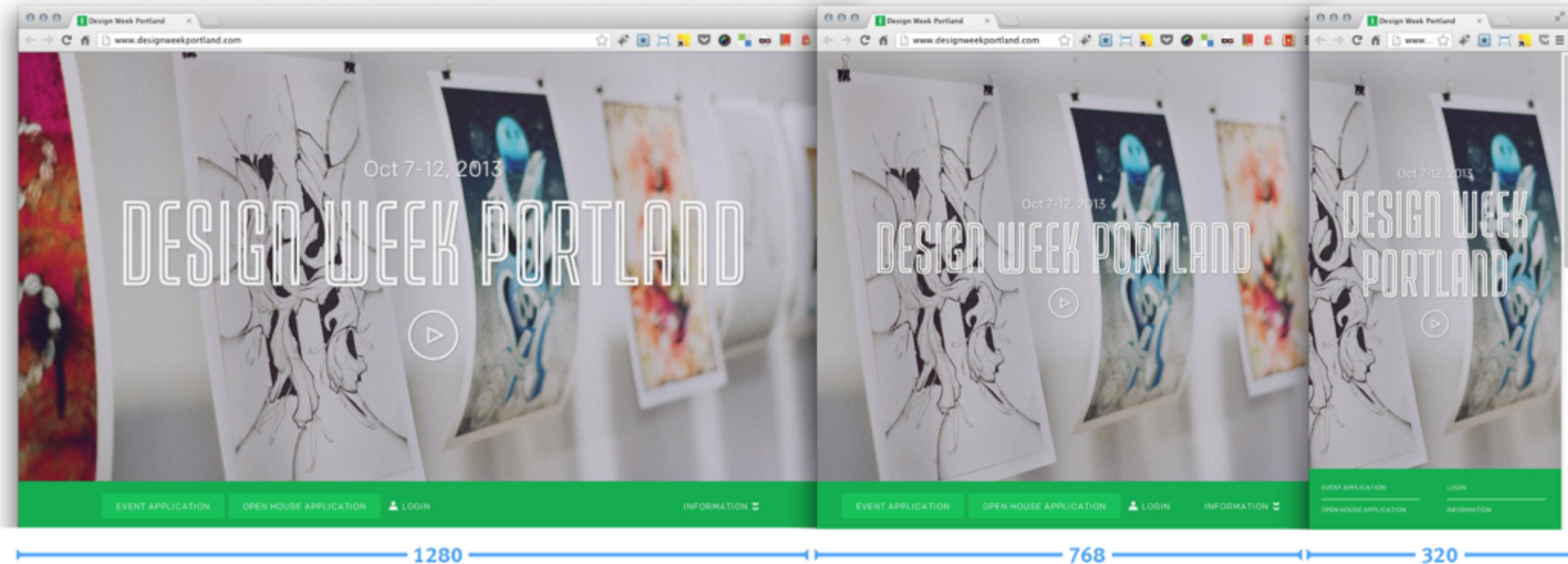


```
.responsive-scale-bg {  
    width: 100%;  
    padding-bottom: 66.6666666667%; /* 960÷1440×100 */  
    background: url(img/fluid/image-1440x960.jpg);  
    background-size: cover;  
}
```

## 재단 이미지

재단 이미지(Crop Image)는 이미지를 포함하는 컨테이너 요소의 폭에 맞춰 크기가 동적으로 잘려지는 이미지를 말한다. (Photoshop의 클리핑 마스크에 가깝다)

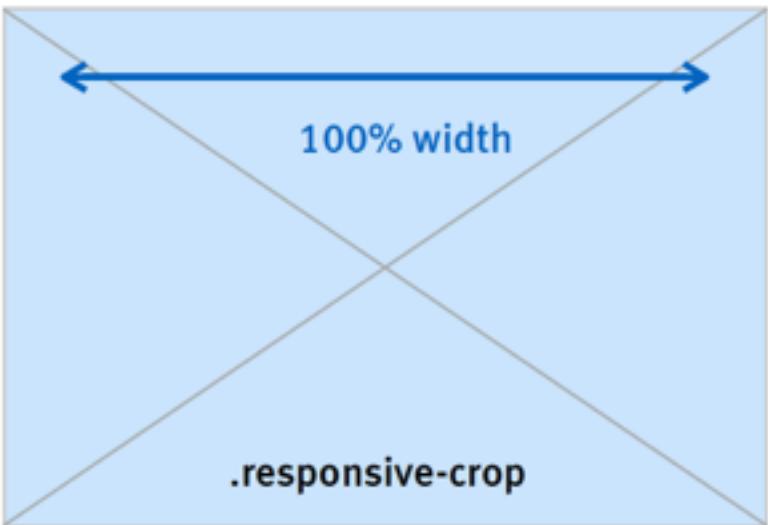
당신의 커리어 전환점 패스트캠퍼스



## 재단 이미지

재단 이미지(Crop Image)는 이미지를 포함하는 컨테이너 요소의 폭에 맞춰 크기가 동적으로 잘려지는 이미지를 말한다. (Photoshop의 클리핑 마스크에 가깝다)

### 배경 이미지



```
.responsive-crop{  
width: 100%;  
height: 960px;  
background-image: url(img/fluid/image-1440x960.jpg) center top;  
background-size: cover;  
}
```

## 재단 이미지

재단 이미지(Crop Image)는 이미지를 포함하는 컨테이너 요소의 폭에 맞춰 크기가 동적으로 잘려지는 이미지를 말한다. (Photoshop의 클리핑 마스크에 가깝다)

콘텐츠 이미지



4:1



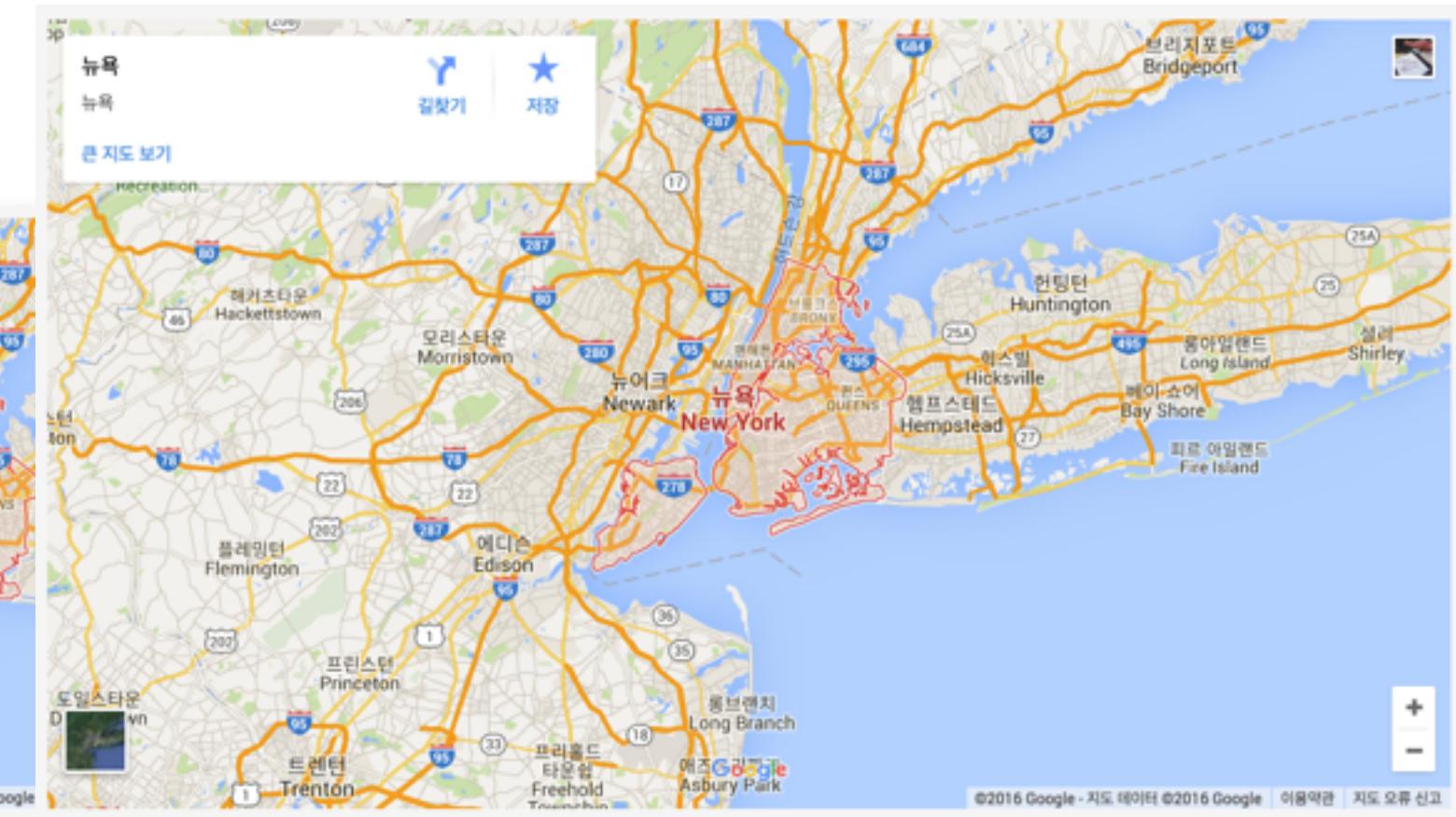
```
.responsive-crop-container {  
    position: relative;  
    height: 120px;  
}
```

```
.responsive-crop-content {  
    position: absolute;  
    left: 50%;  
    width: 100%;  
    height: auto;  
    transform: translateX(-50%);  
}
```

## 유연한 아이프레임

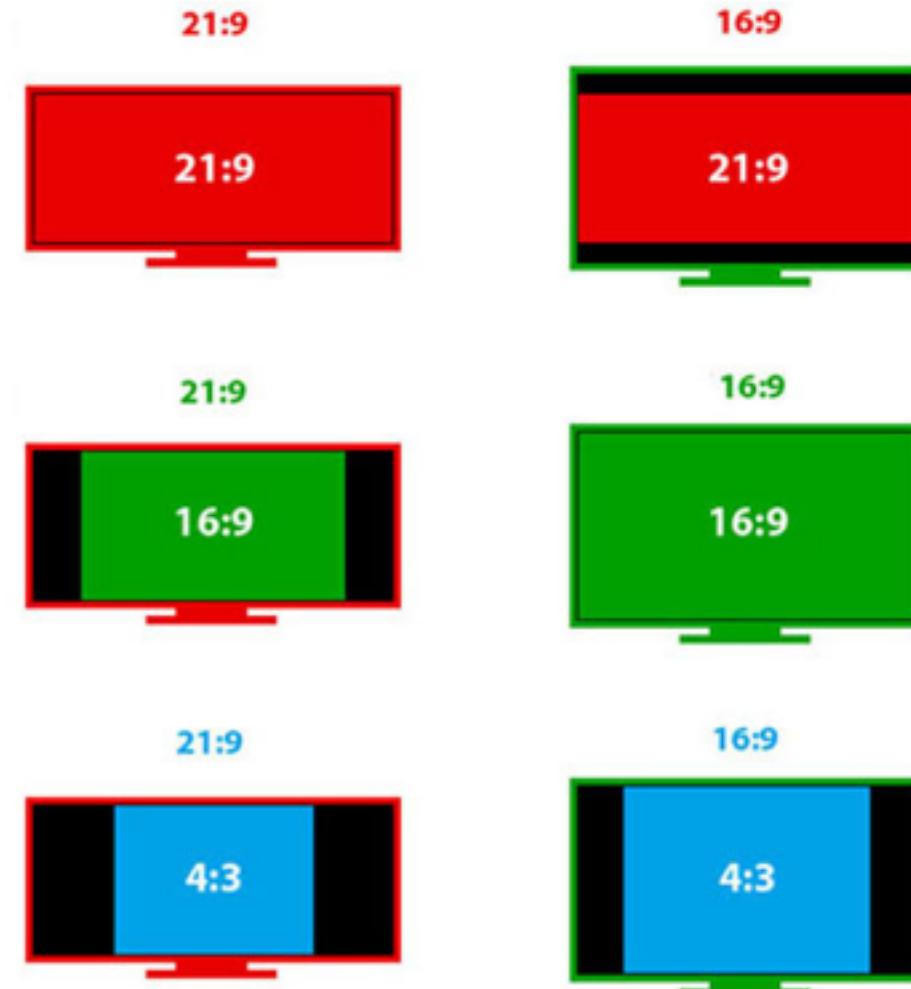
유연한 아이프레임(Flexible iframe)은 아이프레임을 포함하는 컨테이너 요소의 폭에 맞춰 크기가 변경되는 것을 말한다.

당신의 커리어 전환점 퍼스트캠퍼스



## 유연한 아이프레임

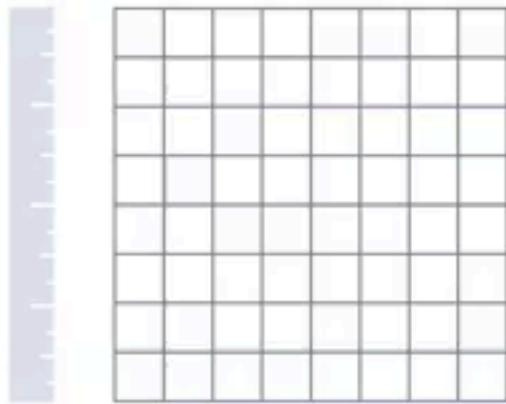
유연한 아이프레임(Flexible iframe)은 아이프레임을 포함하는 컨테이너 요소의 폭에 맞춰 크기가 변경되는 것을 말한다.



```
.responsive-container {  
    position: relative;  
    /* 4:3 = 75%,  
       16:9 => 56.25%,  
       21:9 = 42.857142857 */  
    padding-bottom: 56.25%;  
    height: 0;  
    overflow: hidden;  
    max-width: 100%;  
}  
  
.responsive-container iframe {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
}
```

## 픽셀 밀도

픽셀 밀도란? 공간(대부분 inch에서 사용)에 픽셀이 들어가는 물리적인 수치를 말한다.  
첫 번째 맥킨토시 컴퓨터는 인치당(inch) 72픽셀 이었다. (제록스에 의한 연구 결과)



## 픽셀 밀도

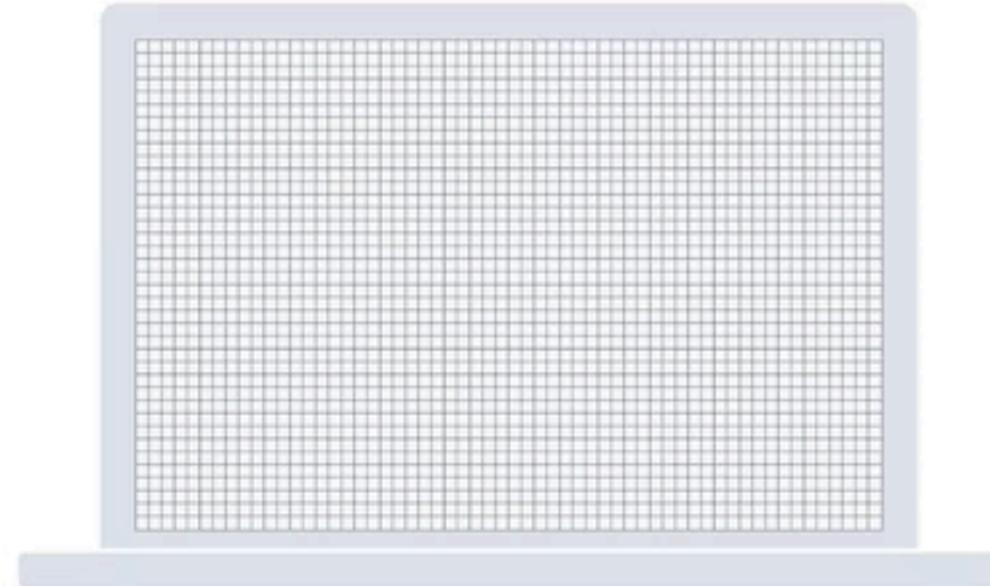
픽셀 밀도란? 공간(대부분 inch에서 사용)에 픽셀이 들어가는 물리적인 수치를 말한다.  
첫 번째 맥킨토시 컴퓨터는 인치당(inch) 72픽셀 이었다. (제록스에 의한 연구 결과)



오리지널 맥 아이콘 디자인 made by Susan Kare.

## 디바이스 픽셀 밀도

애플(Apple)사가 2010년 inch 당 픽셀을 x2 배로 올려 엄청나게 선명한 레티나 디스플레이를 소개한 이후, 디자인(설계) 과정에서 디바이스 픽셀을 고려해야 한다.



## 디바이스 퍽셀 밀도

벡터 그래픽과 달리 비트맵 그래픽은 디바이스의 퍽셀 밀도(Device Pixel Density)에 영향을 받는다. 고해상도를 지원하는 디바이스에 적합한 그래픽 제작이 필요하다.

Pre-Retina Display



Retina Display



메일 아이콘의 봉투와 텍스트 레이블의 선명함이 차이가 난다.



## 디바이스 퍽셀 밀도

벡터 그래픽과 달리 비트맵 그래픽은 디바이스의 퍽셀 밀도(Device Pixel Density)에 영향을 받는다. 고해상도를 지원하는 디바이스에 적합한 그래픽 제작이 필요하다.

The screenshot shows a website for '이롭게' (Iropke) with a yellow circle highlighting the logo. A blue box highlights the developer console output where `window.devicePixelRatio` is shown as 2, indicating a Retina display. The developer tools interface includes tabs for Elements, Console, Sources, Network, Profiles, Timeline, Application, Security, and more.

```
> window.devicePixelRatio
< 2
```

Retina Display

고해상도 디스플레이에서 x1 배율 비트맵 이미지는 뿌옇게 보인다.

Chrome 파일 수정 보기 방문 기록 북마크 사용자 창 도움말

PSD - Google 드라이브 Online Image Compressor Airbnb - 어디서나 NAVER yamoo9

www.naver.com

네이버 앱 다운로드 시스템 글꼴 ?

NAVER 검색

비빔밥 양념 만드는 법 등글레차 효능 금성 A형 간암 증상

메일 카페 블로그 지식IN 쇼핑 TV캐스트 사진 뉴스 증권 부동산 지도 영화 뮤직 책 웹툰 N Pay 더보기 1 어버이연합 186

NAVER doo! 고마운 분들께 마음을 전해보세요! 이웃 가게들이 함께 합니다!

연합뉴스 현대중공업, "인력감축·자산매각" 자구안 제출 네이버뉴스 연예 스포츠 경제 패러디

뉴스스탠드 전체언론사 MY뉴스

vNet Korea 05.12 12:25 편집 × B블로터 05.12 11:47 편집 ×

- 스마트폰 시장 더 악화된다는 '3대 조짐'
- 산적한 ICT 현안, 국회 미방위 누가 올까
- 5月 보급형 스마트폰 전쟁 불 붙었다
- 'LG G5' 카메라 UX의 5가지 매력
- IT 덕분에 재발견한 '옛 노래의 즐거움'
- "아이폰7+, 듀얼 카메라+3GB 램 장착"
- "네이버에서 만나보셨나요? 인공지능 채팅 로봇"
- [IT일상] 스마트 모빌리티
- 교육용 협업 도구 '프레시그레이드', 130억원 투자 유치
- MS, 에저 클라우드용 데이터센터 부산에 짓는다
- "페이스북이 보수적인 뉴스 고의로 배제했다"
- 오픈소스 취약점 단번에 찾아주는 '소스클리어'

구독설정 1/3 < >

라이프 스포츠 차/테크 웹툰 게임 TV/동영상 뮤직 영화 책/문학 자식/교양 오픈캐스트 공익/나눔

쇼핑 상품 쇼핑몰 MEN

THE 보이는 5M - 2관 태양의 연인 민효린, 연예 비하인드 손여은, 조재현에 치명적 유혹

AEROFLOT 모스크바를 거쳐 유럽으로!

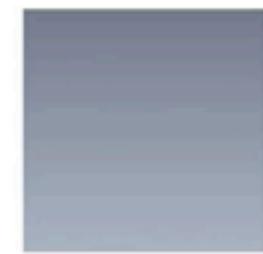
GO마켓 쿠팡 옥션 11번가 위메프 티몬 이마트몰 롯데몰 GS샵 롯데닷컴 신세계몰 CJmall 현대마트

Fast camp

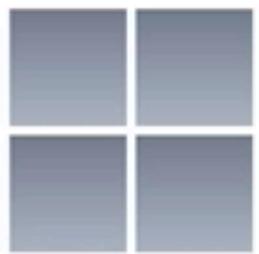
## 디바이스 별 UI 크기



물리적으로 동일한 UI 크기를 유지하려면 픽셀 면적이 x2배일 때, 44px 크기 버튼은 88px이 되어야 한다. 각기 다른 디바이스에 이와같은 UI 개념을 적용하기 위해 디자이너는 원래 x1 크기 제작은 물론 x2 크기 제작이 필요해졌다.



1x



2x

## 디바이스 별 UI 크기



수치 측정 단위 중 픽셀 밀도를 측정하는 독립적인 단위가 없어 디자인 제작과정이 곤란  
스러워졌다. 이에 대한 해결책을 애플은 “Point(pt)”를 제시했다. (웹 UI와는 별개입니다)

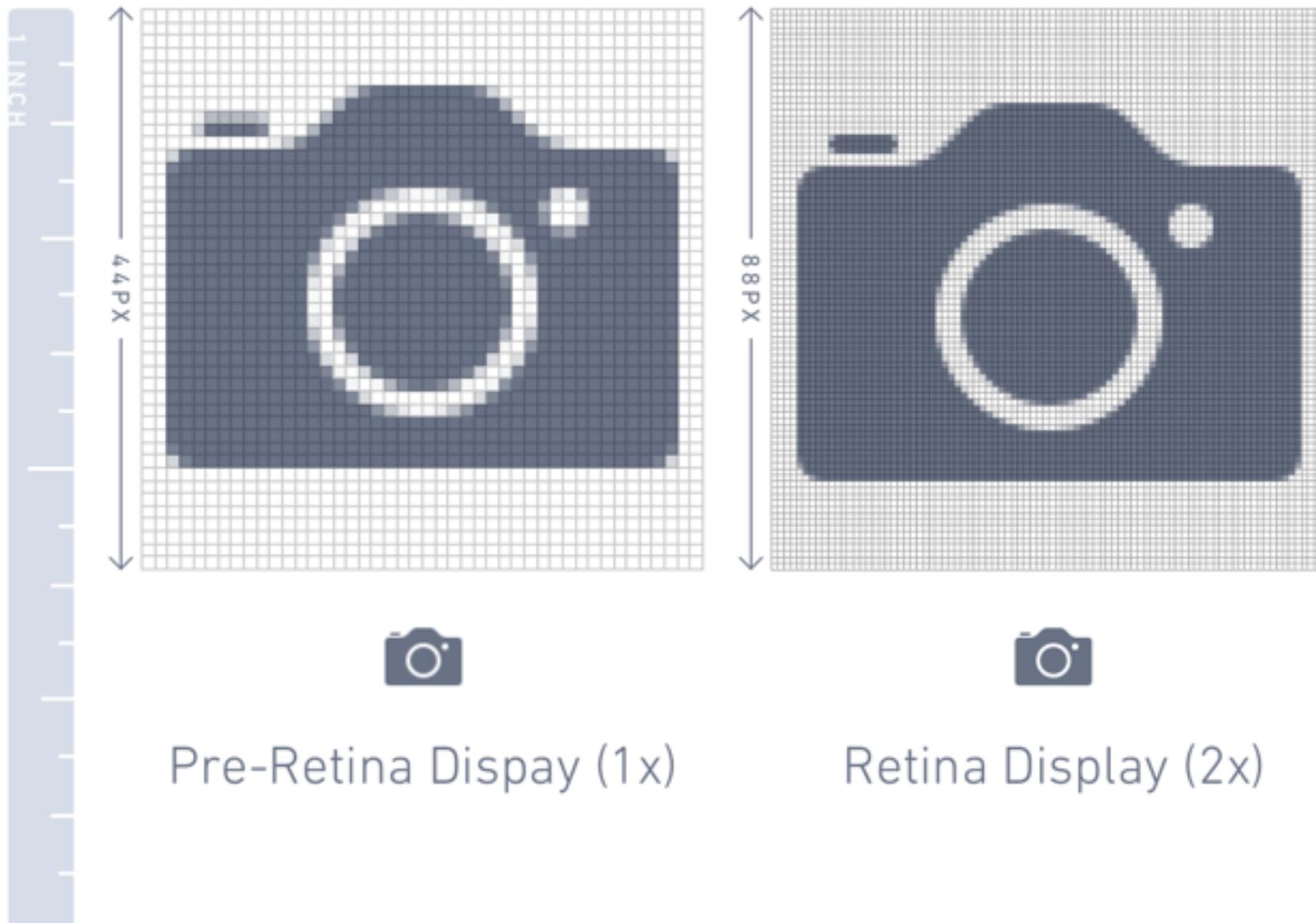


Points

## 디바이스 별 UI 크기



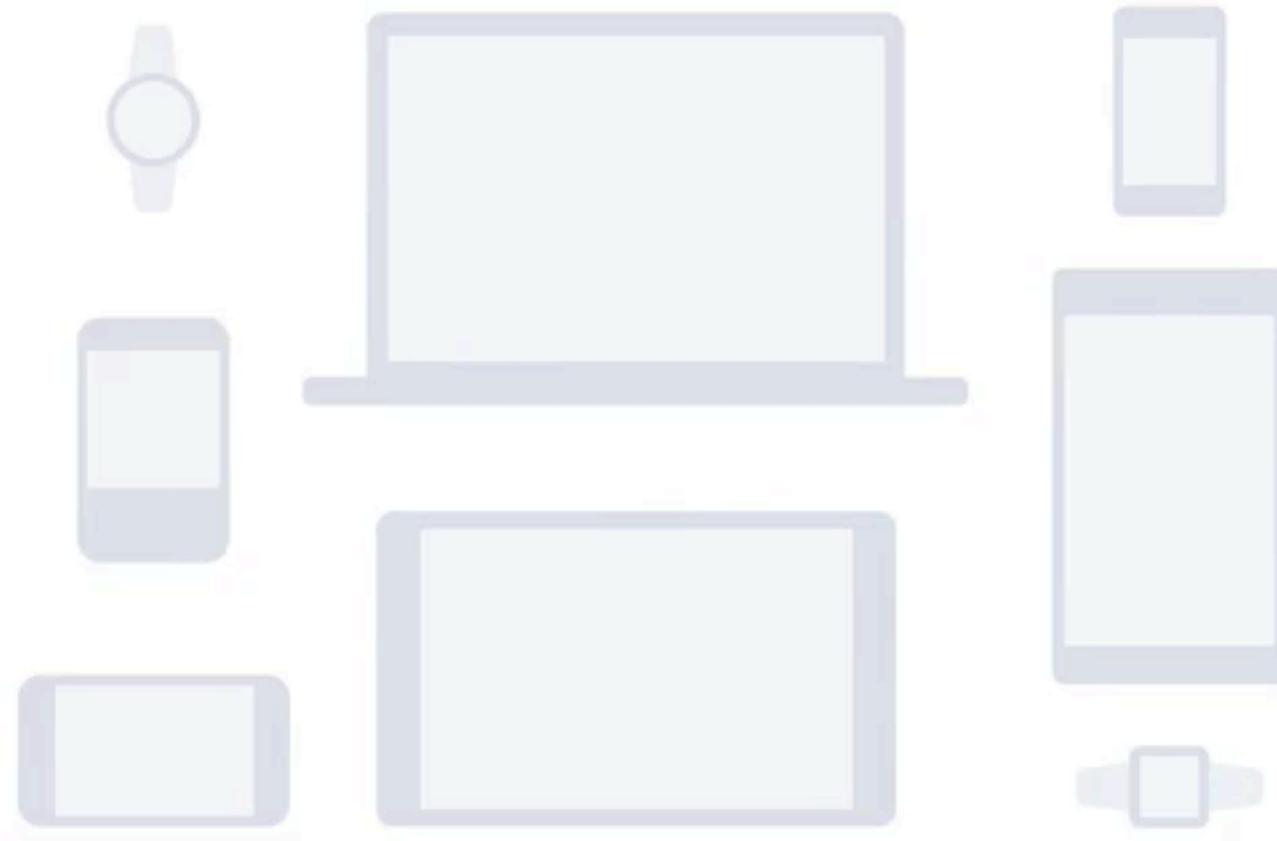
수치 측정 단위 중 픽셀 밀도를 측정하는 독립적인 단위가 없어 디자인 제작과정이 곤란 스러워졌다. 이에 대한 해결책을 애플은 “Point(pt)”를 제시했다. (웹 UI와는 별개입니다)



## 디바이스 별 UI 크기



애플과 달리 안드로이드 진영은 장치 독립적인 픽셀 밀도(DIP) 단위인 DP를 만들어낸다.  
문제는 정수 배율을 가진 애플과 달리 실수 배열을 가졌다는 점이다. (웹 UI와는 별개입니다)

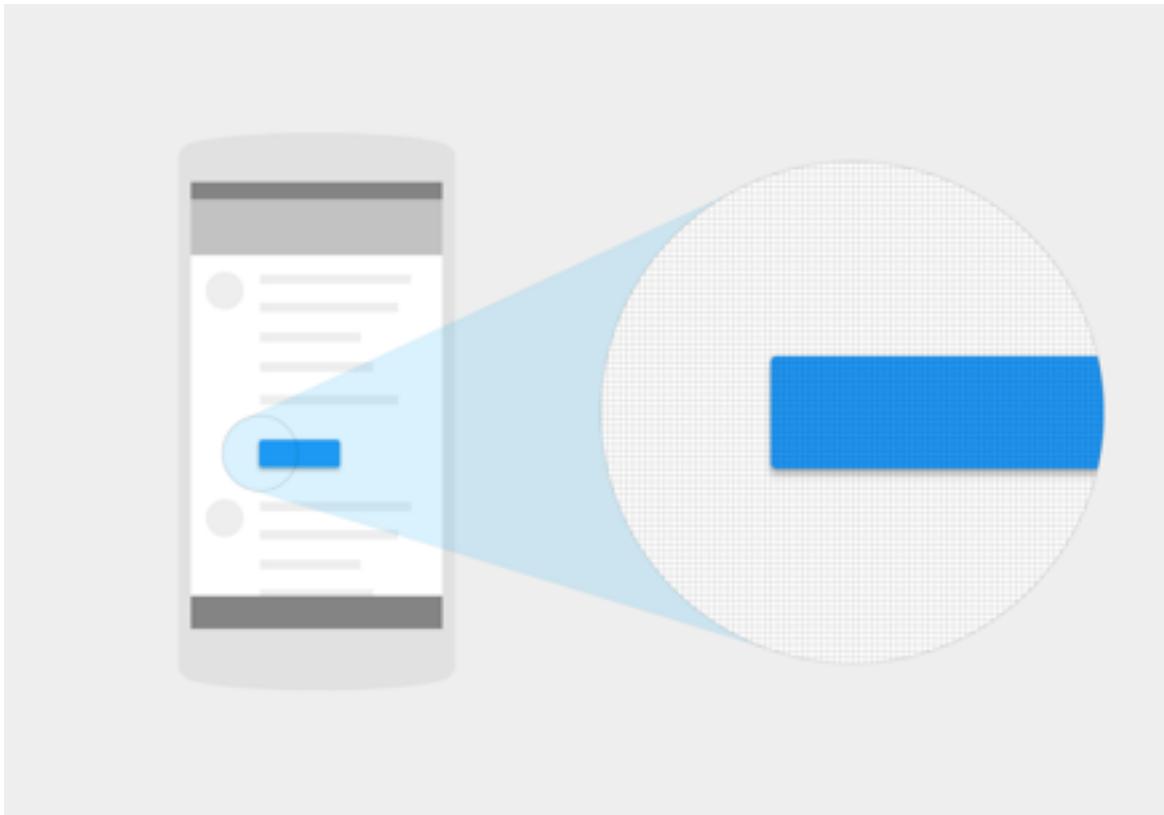


## 디바이스 별 UI 크기

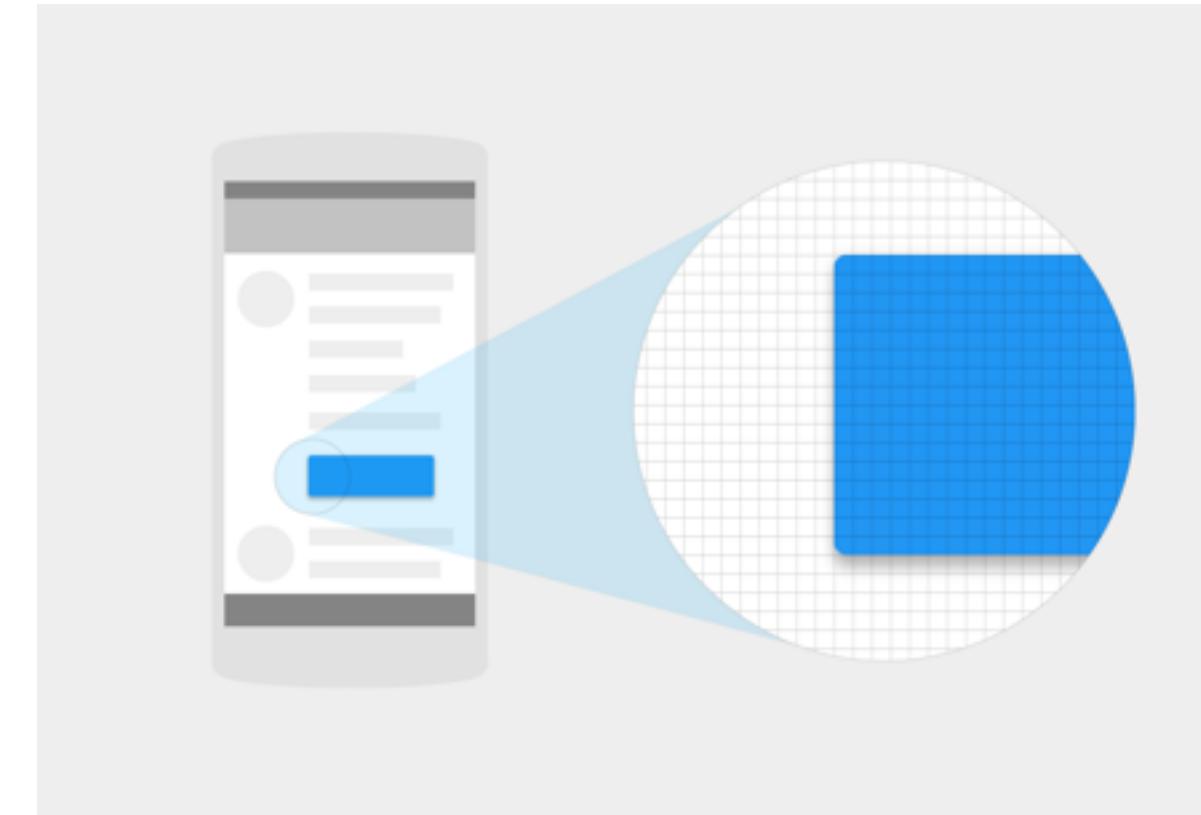


애플과 달리 안드로이드 진영은 장치 독립적인 픽셀 밀도(DIP) 단위인 DP를 만들어낸다.  
문제는 정수 배율을 가진 애플과 달리 실수 배율을 가졌다는 점이다. (웹 UI와는 별개입니다)

당신의 커리어 전환점 패스트캠퍼스



High-density screen

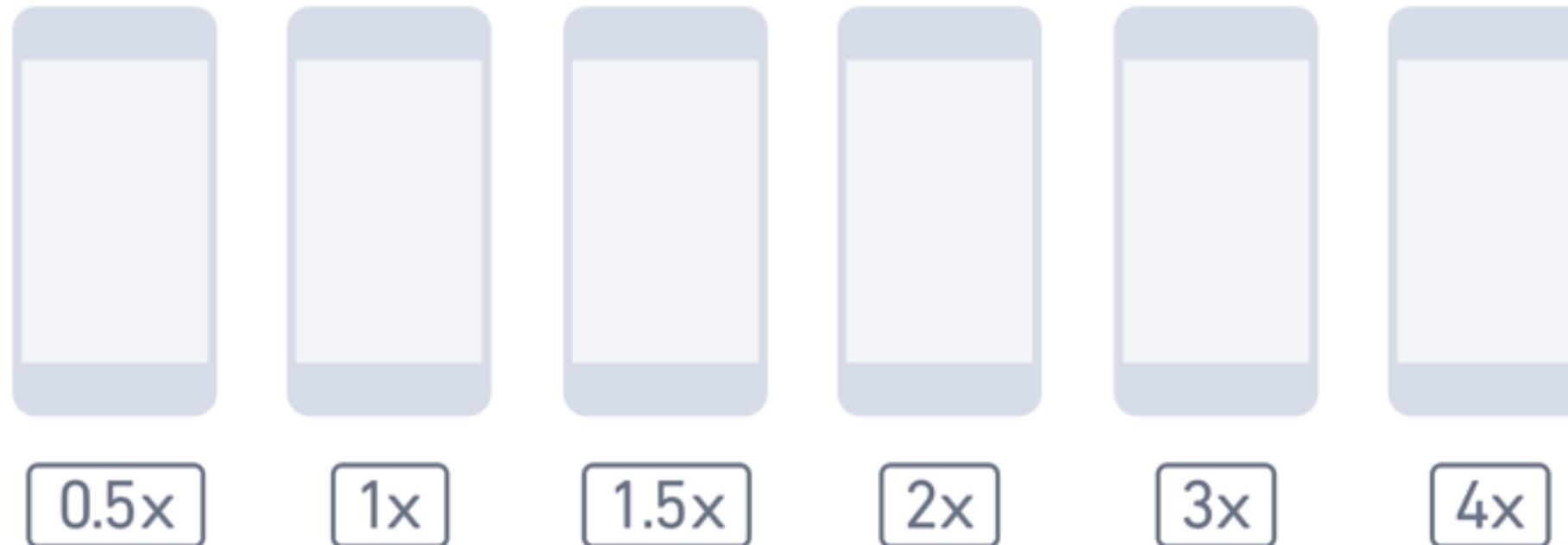


Low-density screen

## 디바이스 별 UI 크기



애플과 달리 안드로이드 진영은 장치 독립적인 픽셀 밀도(DIP) 단위인 DP를 만들어낸다.  
문제는 정수 배율을 가진 애플과 달리 실수 배율을 가졌다는 점이다. (웹 UI와는 별개입니다)



## 디바이스 별 UI 크기



애플과 달리 안드로이드 진영은 장치 독립적인 픽셀 밀도(DIP) 단위인 DP를 만들어낸다.  
문제는 정수 배율을 가진 애플과 달리 실수 배율을 가졌다는 점이다. (웹 UI와는 별개입니다)

당신의 커리어 전환점 패스트캠퍼스

Screen resolution	dpi	Pixel ratio	Image size (pixels)
xxxhdpi	640	4.0	400 x 400
xxhdpi	480	3.0	300 x 300
xhdpi	320	2.0	200 x 200
hdpi	240	1.5	150 x 150
mdpi	160	1.0	100 x 100

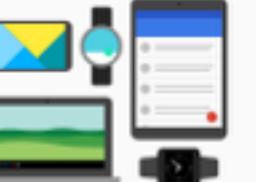
## 디바이스 별 UI 크기 정보

워치/모바일/태블릿/데스크톱 등  
디바이스의 규격 정보를 자세하게 제공한다.

### Device metrics

Type	Device	Platform	Screen dimensions <small>in cm</small>	Aspect Ratio	Width × Height <small>dp</small>	Width × Height <small>px</small>	Density
□	Android One	Android	4.5 in × 2.2 × 3.9 in	16 : 9	320 × 569 dp	480 × 854 px	1.5 hdpi
⌚	Asus Zen Watch	Android	1.6 in × 1.2 × 1.2 in	1 : 1	213 × 213 dp	328 × 328 px	1.5 hdpi
□	Dell Venue 8	Android	8.4 in × 4.5 × 7.1 in	16 : 10	800 × 1280 dp	1600 × 2560 px	2.0 xhdpi
□	HTC One M8	Android	5.0 in × 2.5 × 4.4 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
□	HTC One M9	Android	5.0 in × 2.5 × 4.4 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
⌚	LG G Watch	Android	1.7 in × 1.2 × 1.2 in	1 : 1	187 × 187 dp	288 × 288 px	1.5 hdpi
⌚	LG G Watch R	Android	1.8 in × 1.3 × 1.3 in	1 : 1	213 × 213 dp	328 × 328 px	1.5 hdpi
□	LG G2	Android	5.2 in × 2.5 × 4.5 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
□	LG G3	Android	5.5 in × 2.7 × 4.8 in	16 : 9	480 × 853 dp	1440 × 2560 px	3.0 xxhdpi

**Information**



**Material metrics**

Select a device to view detailed information, recommended material design measurements, and values for portrait and landscape orientations.

## 밀도 변환기

픽셀(px) 값을 입력하면 각 디바이스 밀도에 해당되는  
픽셀 값을 반환하는 도구이다.

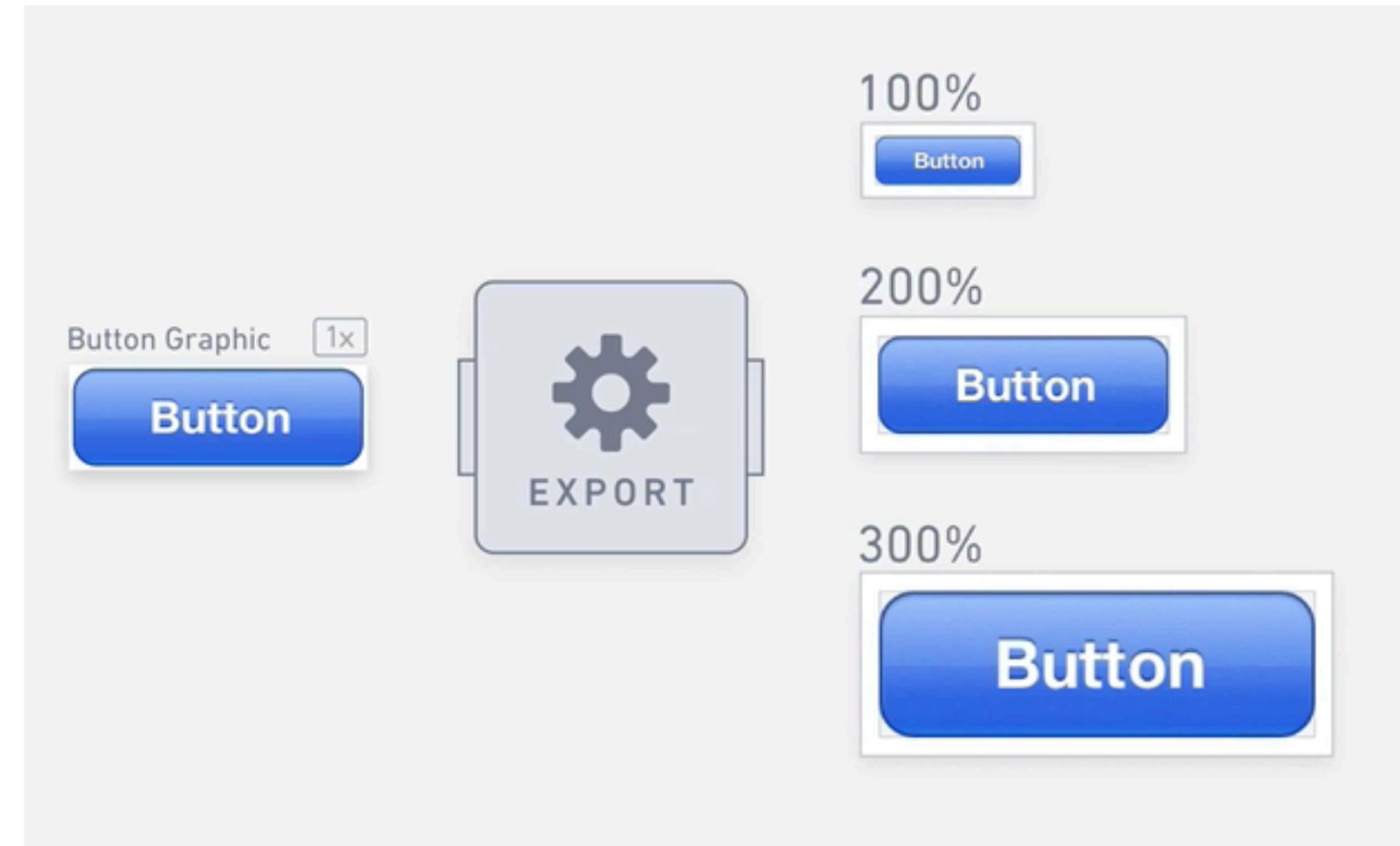
### DENSITY CONVERTER *by Jeff Broderick*

<b>LDPI 0.75X</b>	12 PX	18 PX
<b>BASE / MDPI 1X</b>	16 PX	24 PX
<b>TDPI 1.33X</b>	22 PX	32 PX
<b>HDPI 1.5X</b>	24 PX	36 PX
● <b>RETINA / XHDPI 2X</b>	32 PX	48 PX
<b>XXHDPI 3X</b>	48 PX	72 PX
<b>XXXHDPI 4X</b>	64 PX	96 PX

● *Indicates your screen's current density.*

## 디바이스 픽셀 밀도 UI 그래픽 내보내기

당신의 커리어 전환점 패스트캠퍼스

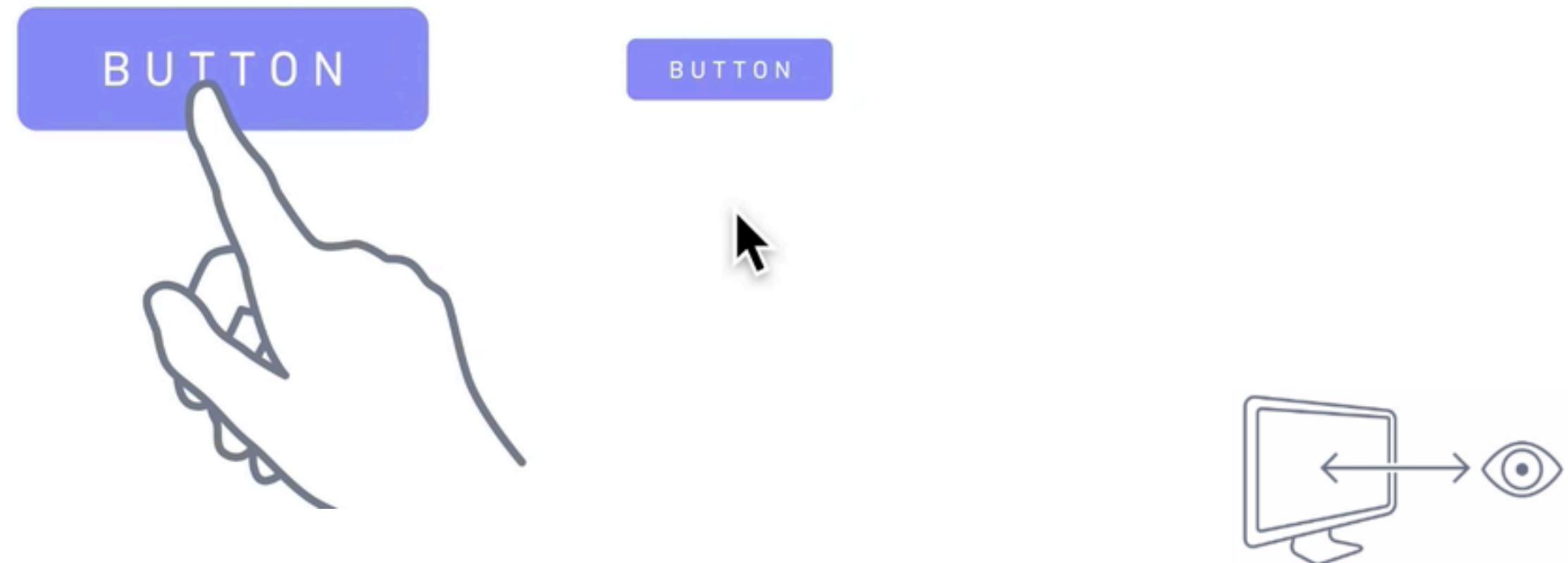


포토샵을 사용하는 그래픽 디자이너는 x1 배율의 벡터 그래픽을 제작한 후,  
디바이스 픽셀 밀도를 고려하여 내보내야 한다. (비트맵의 경우는 스마트오브젝트를 사용한다)

## 사람이 보는 크기의 인식 고려

태블릿에서의 앱 아이콘은 폰보다 크기가 커야 한다. 그렇게 하기 위한 2가지 방법 중 하나는 픽셀 밀도는 낮추는 것이고, 다른 하나는 버튼 크기를 조정하는 것이다.

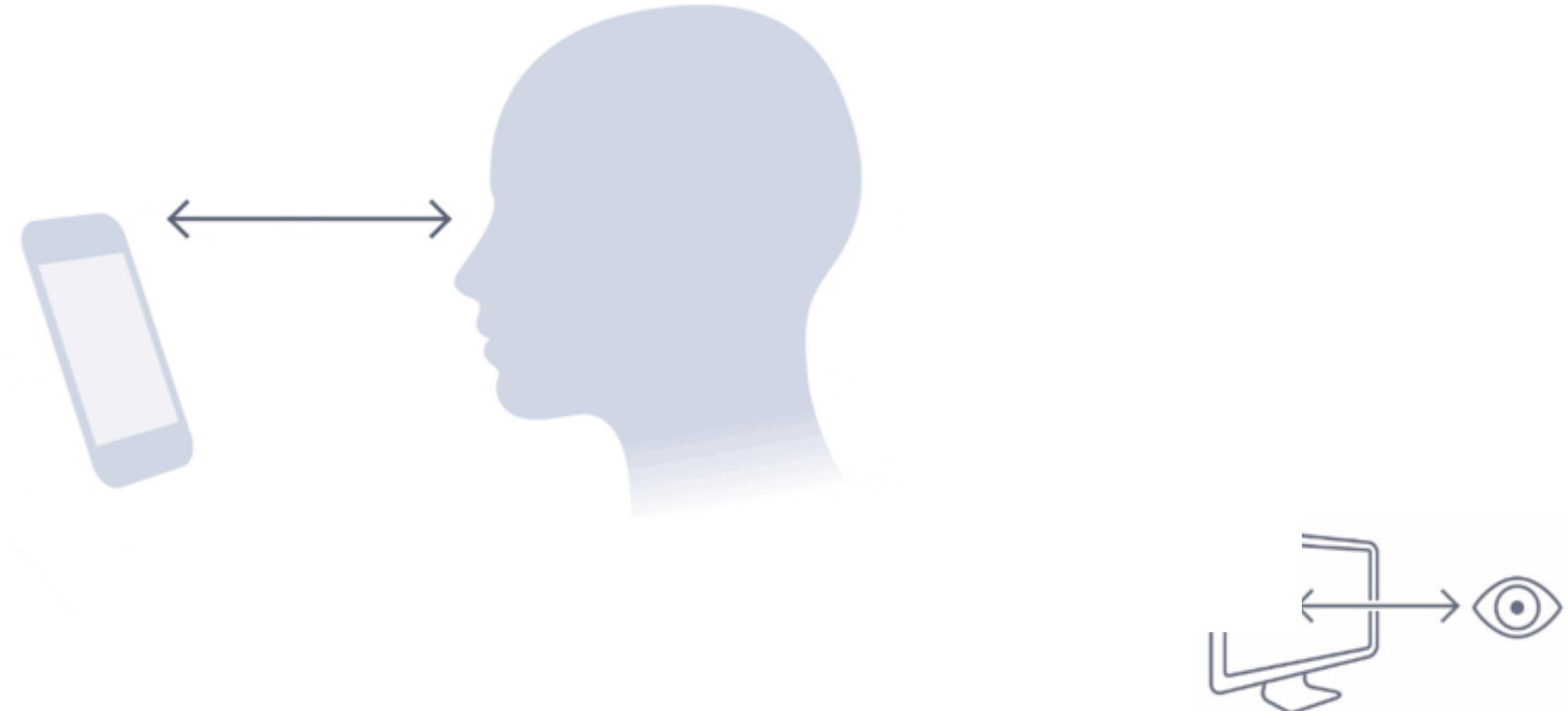
당신의 커리어 전환점 패스트캠퍼스



## 사람이 보는 크기의 인식 고려

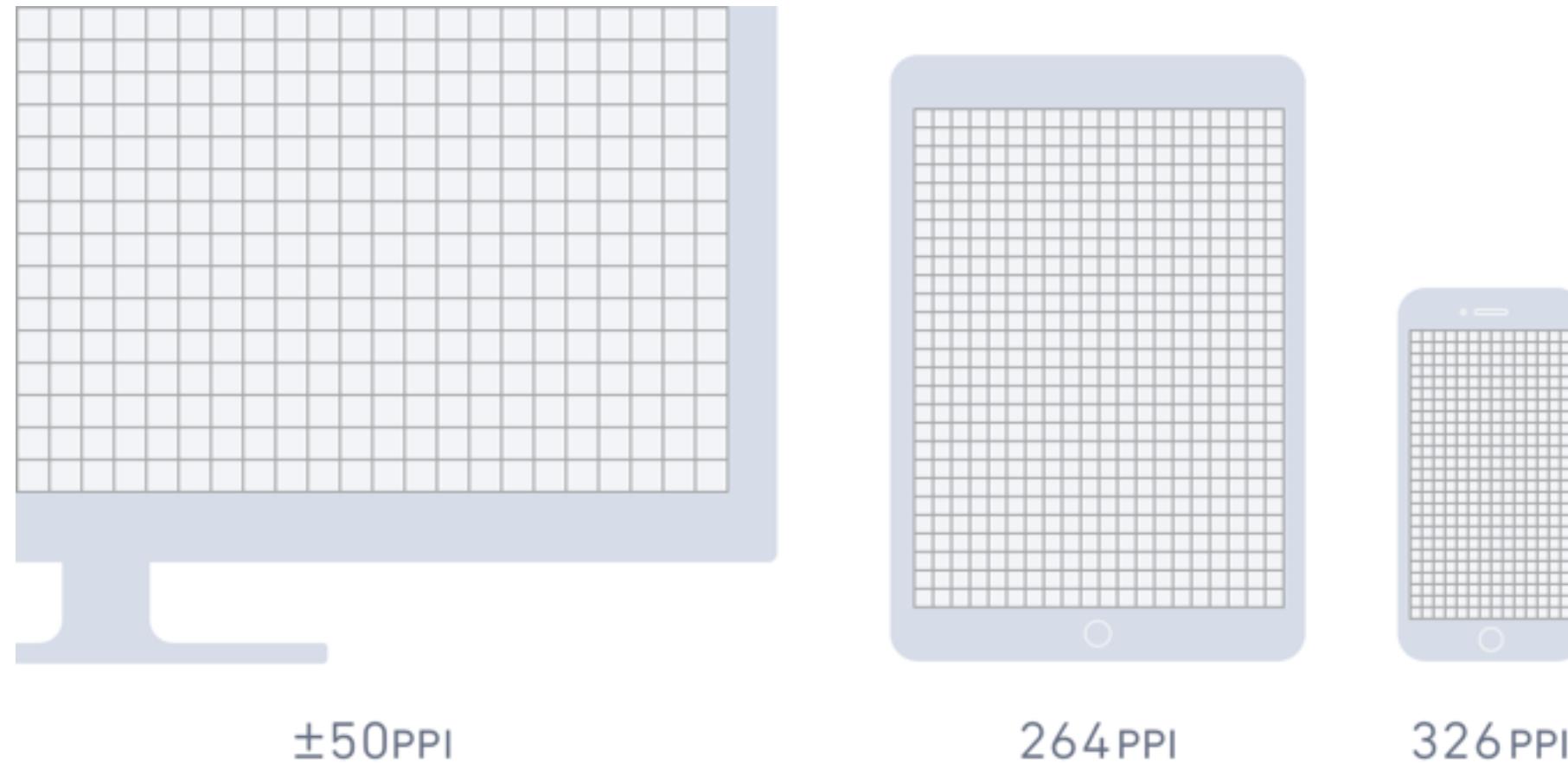
태블릿에서의 앱 아이콘은 폰보다 크기가 커야 한다. 그렇게 하기 위한 2가지 방법 중 하나는 픽셀 밀도는 낮추는 것이고, 다른 하나는 버튼 크기를 조정하는 것이다.

당신의 커리어 전환점 패스트캠퍼스



## 1. 픽셀 밀도 낮추기

대형 스크린은 멀리서 보기에 보통 낮은 픽셀 밀도를 가지고 있다. 일반적으로 TV는 생각보다 아주 낮은 인치당 40픽셀을 가지고 있는데 반해, 아이폰 레티나는 326ppi로 매우 높은 픽셀 밀도를, 아이패드 레티나는 대략 226ppi 정도의 픽셀 밀도를 가지고 있다. 아이패드의 큰 픽셀(스크린은 덜 빽빽함)이 인터페이스를 약간 크게 보이게 해준다.



## 2. 크기 바꾸기

픽셀 밀도를 낮추는 것만으로는 충분하지 않아 특정 디자인 요소들은 큰 크기로 만들어야 하기도 한다. 아이패드 앱 아이콘에서 이러한 경우가 발생하는데 아이폰에서는 앱 아이콘이 60x60 크기인데 아이패드의 큰 스크린은 더 많은 공간을 필요로 하기에 76x76 크기의 앱 아이콘을 사용하여 시각적인 크기 개선을 해야 한다.



60 PT

326 PPI  
iPhone Display



60 PT

264 PPI  
iPad Display

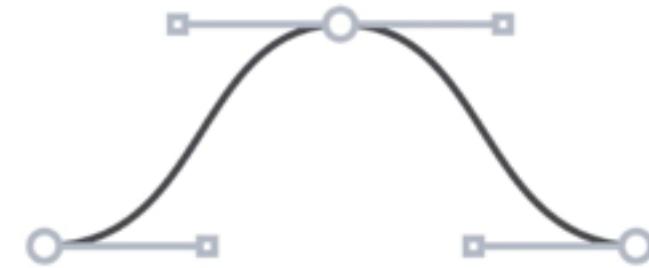


76 PT

264 PPI  
iPad Display

## 디자인 시작점

사진 같은 비트맵 이미지를 제외하고는 대부분의 UI는 벡터를 사용하여 제작하며 디자인 배율은 x1에서 시작해야 한다. (비트맵 이미지는 스마트 오브젝트를 사용하여 내보낸다)



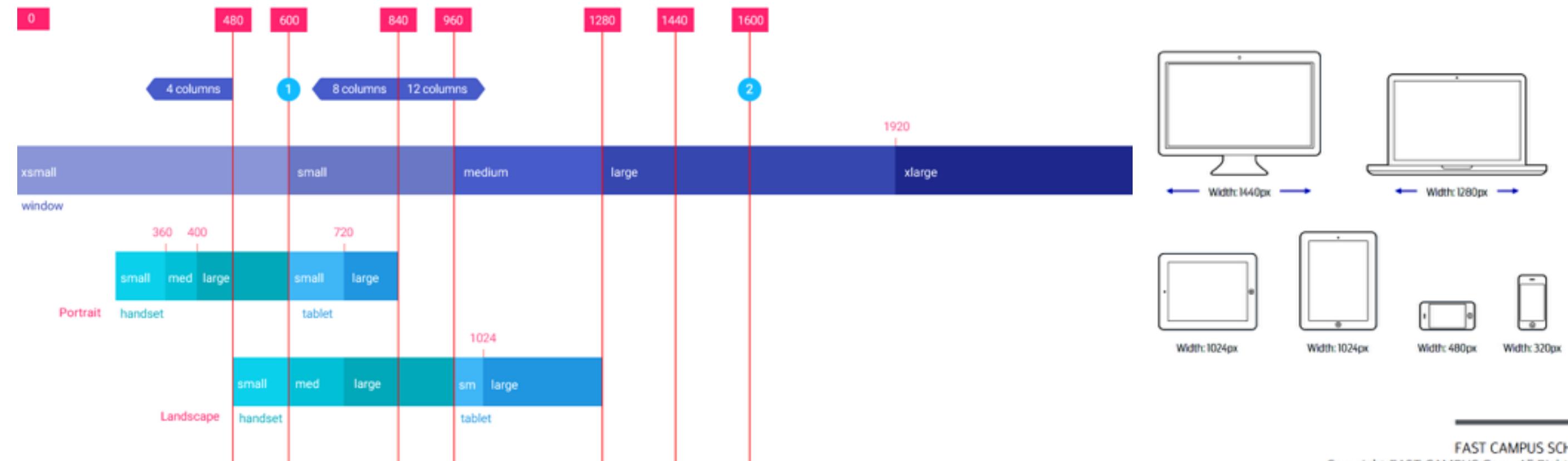
Design in Vector



Design at 1x

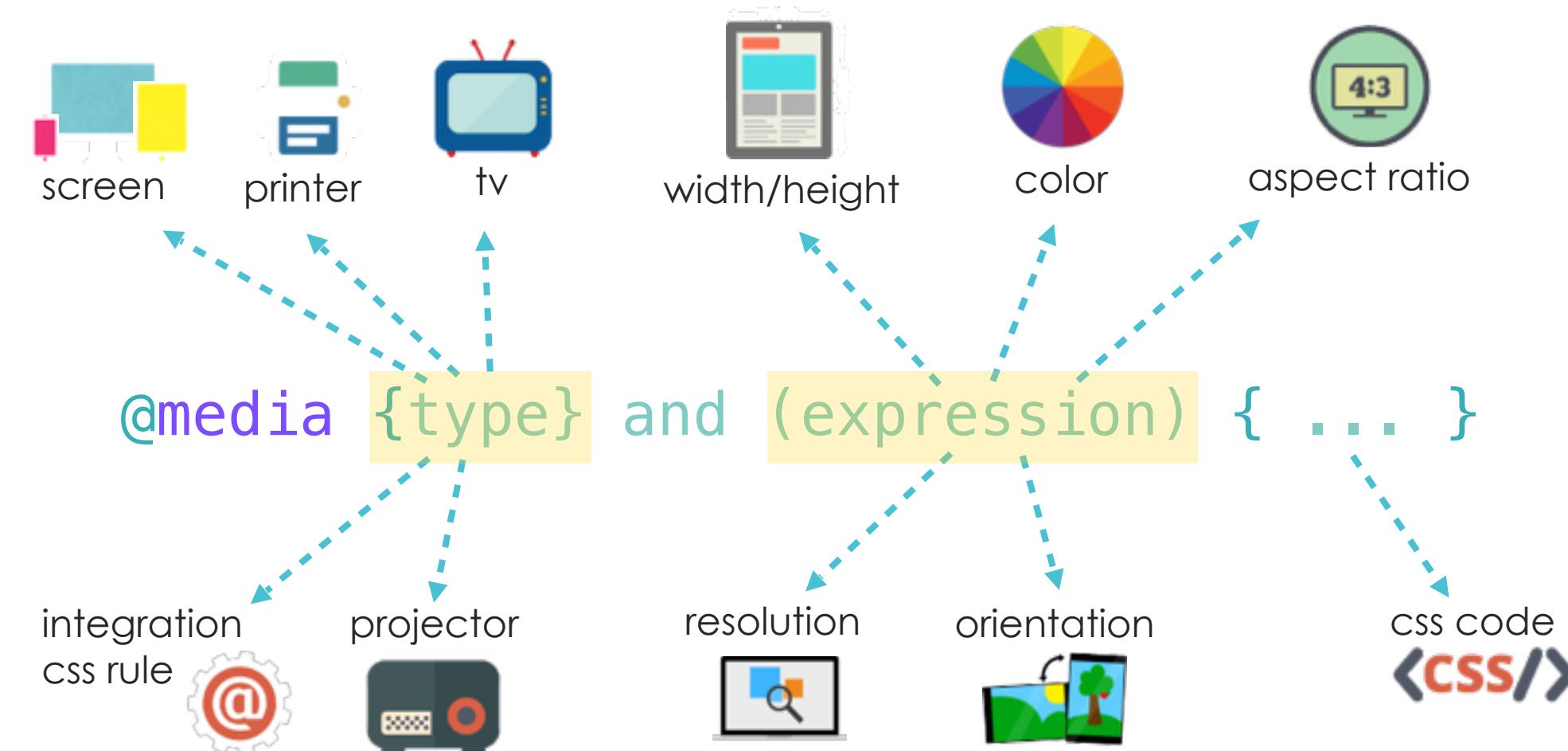
## 브레이크포인트

RWD 프로젝트 진행 전, 주요 사용자 층이 사용하는 디바이스 환경을 분석한 후, 최적화되고 유효한 중단점(Breakpoint)을 설계(Design)를 한다.



## CSS3 미디어쿼리

미디어쿼리는 각 디바이스 환경을 식별하는 조건 처리 구문으로 CSS3에서 정식 지원한다. 이를 사용하여 설계된 중단점에 맞는 최적화된 뷰 디자인을 구현할 수 있다.



# CSS3 미디어쿼리

W3C 표준 미디어쿼리 사양 (Media Queries Recommendation)

Feature	Value	Min/Max	Description
<b>width</b>	length	yes	Display width
<b>height</b>	length	yes	Display height
<b>device-width</b>	length	yes	Device width
<b>device-height</b>	length	yes	Device height
<b>orientation</b>	portrait or landscape	no	Device orientation
<b>aspect-ratio</b>	ratio (w/h)	yes	Ratio of width to height
<b>device-aspect-ratio</b>	ratio (w/h)	yes	Ratio of device-width to device-height
<b>color</b>	integer	yes	Number of bits per color component
<b>color-index</b>	integer	yes	Number of entries in the output device's color lookup table
<b>monochrome</b>	integer	yes	Number of bits per pixel in the monochrome frame buffer
<b>resolution</b>	resolution	yes	Density of pixels of output device, (integer followed by dpi or dpcm)

## CSS3 미디어쿼리 디바이스 가로/세로 폭 길이 감지

```
/* width */
@media screen and (max-width: 600px) { ... }
@media screen and (min-width:200px) and (max-width:400px) { ... }

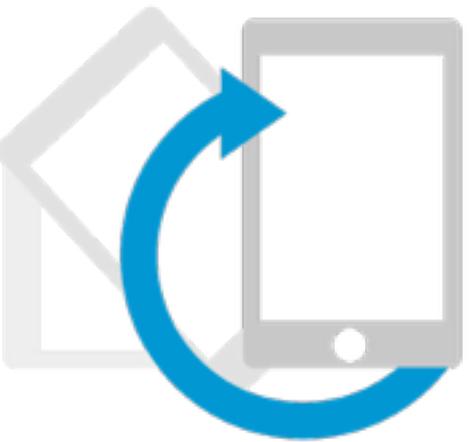
/* height */
@media screen and (max-height:768px) { ... }
@media (min-height:500px) and (max-height:580px) { ... }
```



## CSS3 미디어쿼리 디바이스 회전 방향 감지

당신의 커리어 전환점 패스트캠퍼스

```
/* portrait */  
@media screen and (orientation: portrait) { ... }  
  
/* landscape */  
@media screen and (orientation: landscape) { ... }
```



## CSS3 미디어쿼리 디바이스 픽셀 밀도 감지

당신의 커리어 전환점 패스트캠퍼스

```
/* 300 DPI */
@media print and (min-resolution: 300dpi) { ... }

/* x2 Device Pixel Density */
@media screen and (min-resolution: 2dppx) { ... }
```



## CSS3 미디어쿼리 논리 연산자 응용

```
/* AND */  
@media all and (color) { ... }  
  
/* NOT + AND */  
@media not screen and (color) { ... }  
  
/* ONLY + AND */  
@media only screen and (orientation: portrait) { ... }  
  
/* COMMA */  
@media all and (orientation: landscape),  
      all and (min-width: 480px) { ... }
```



## CSS3 미디어쿼리 호환성

2016년 현재 미디어쿼리는 웹 표준으로 대부분의 웹 브라우저에서 사용 가능하다.  
단, dppx 단위의 경우는 Edge, Chrome, Firefix, Opera Android 만 지원하고 있다.

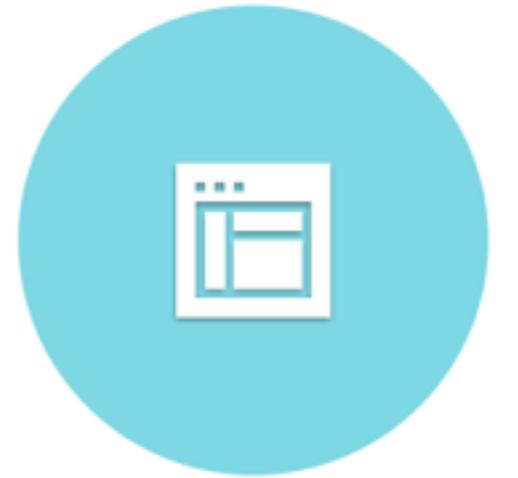




기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

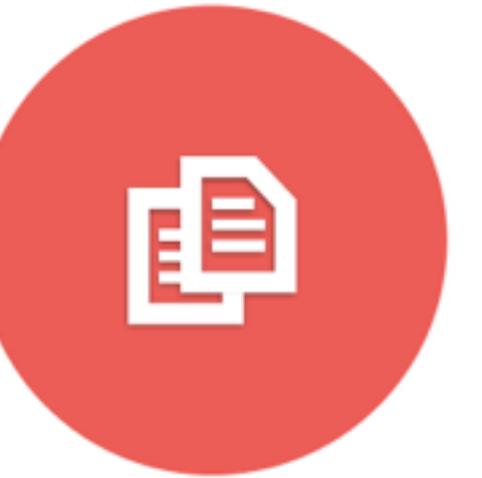
기술을 이용해, 동료를 이용해, 세상을 이용해



Layout



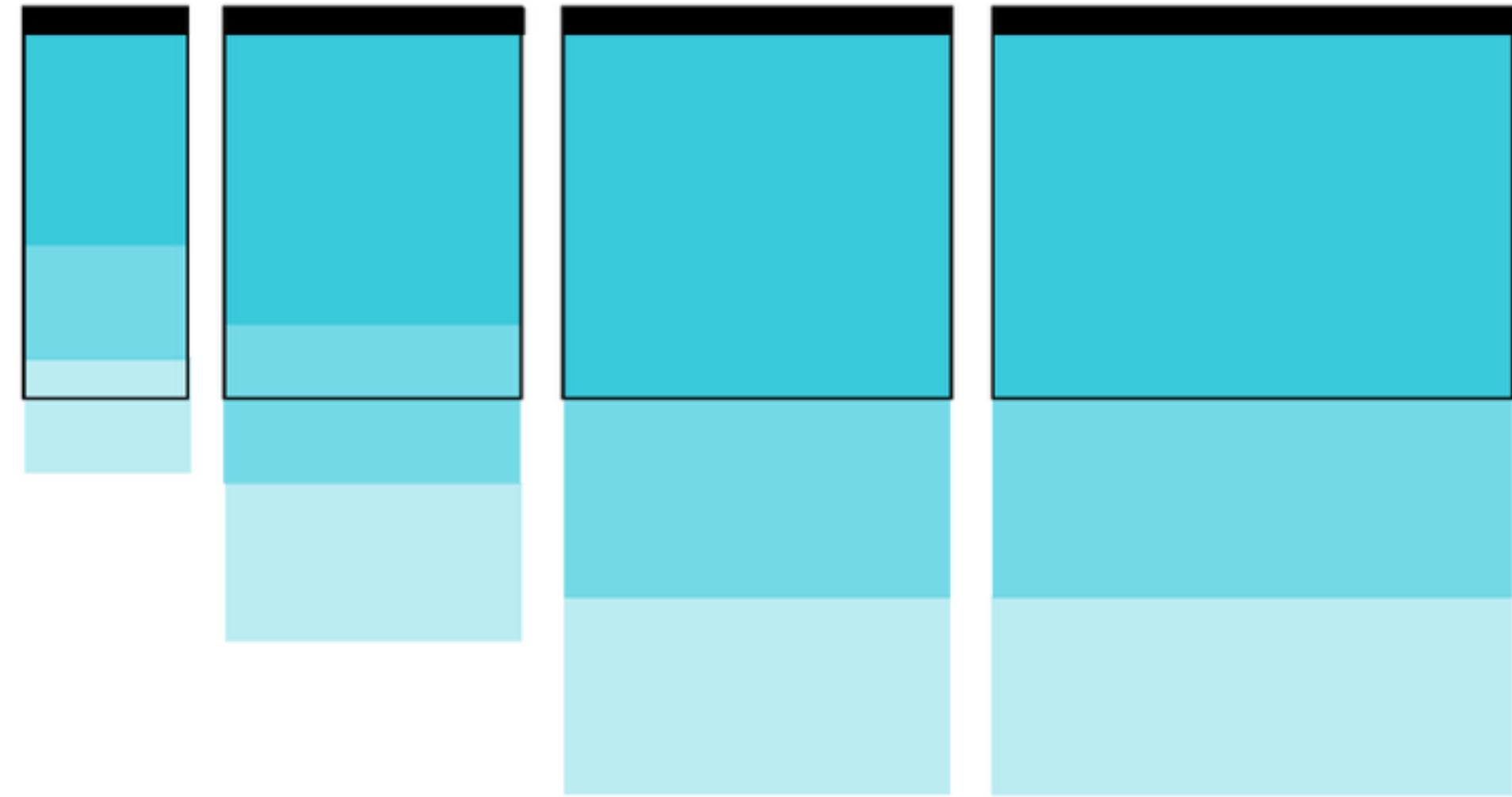
Navigation



Content

## Tiny Tweaks 웹 서비스의 구조는 그대로 유지하면서 스케일만 변경하는 패턴이다.

기술을 이용해, 동료를 이용해, 세상을 이용해



IROPKE Front-End Design CAMPUS



기술을 이용해, 동료를 이용해, 세상을 이용해

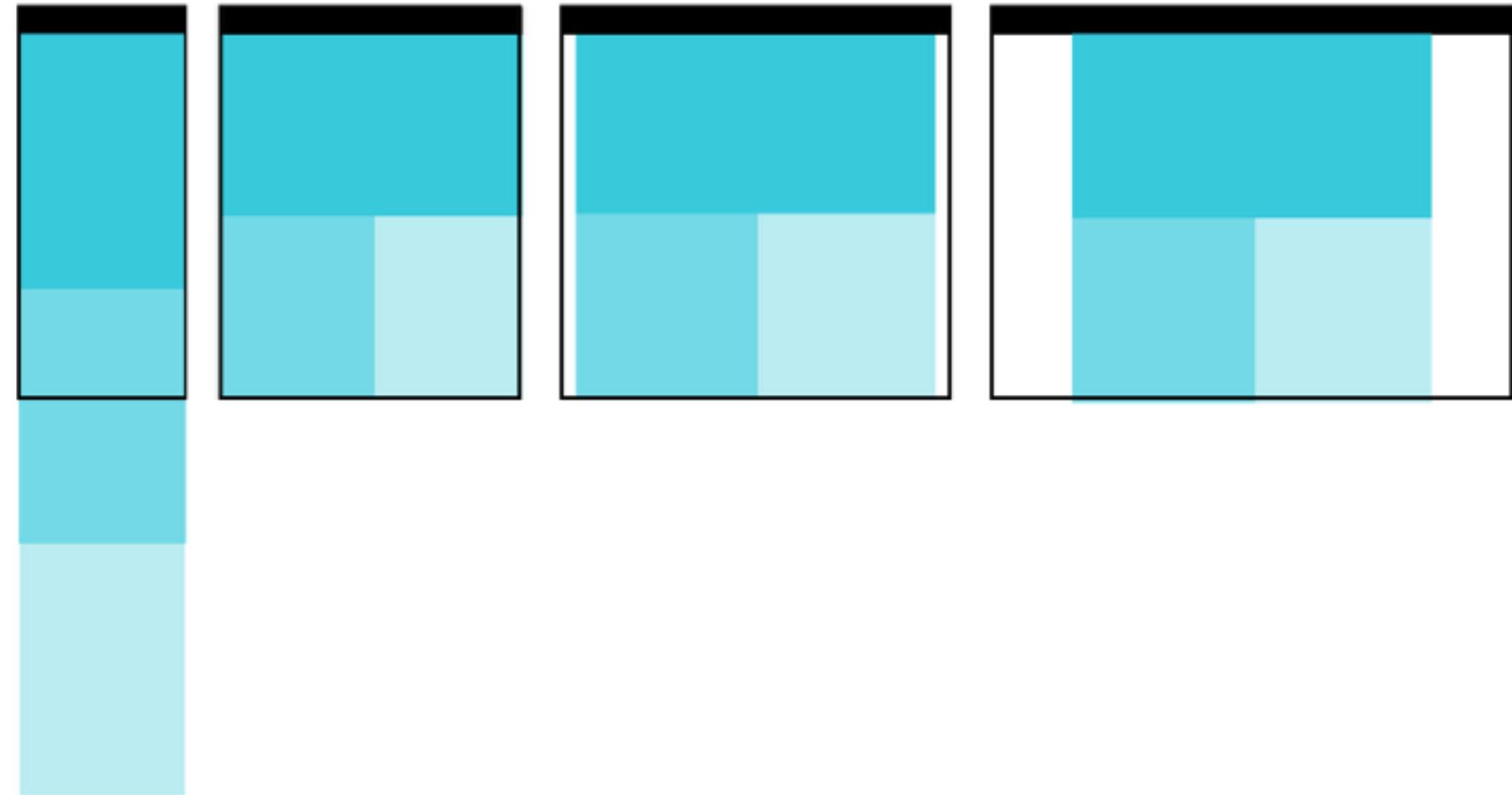


<http://build.microsoft.com/>



## Mostly Fluid

유연한 멀티 컬럼 그리드 레이아웃에 기초하여 화면 크기에 따라 콘텐츠가 수직 방향으로 쌓이게 되는 패턴이다.



기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS





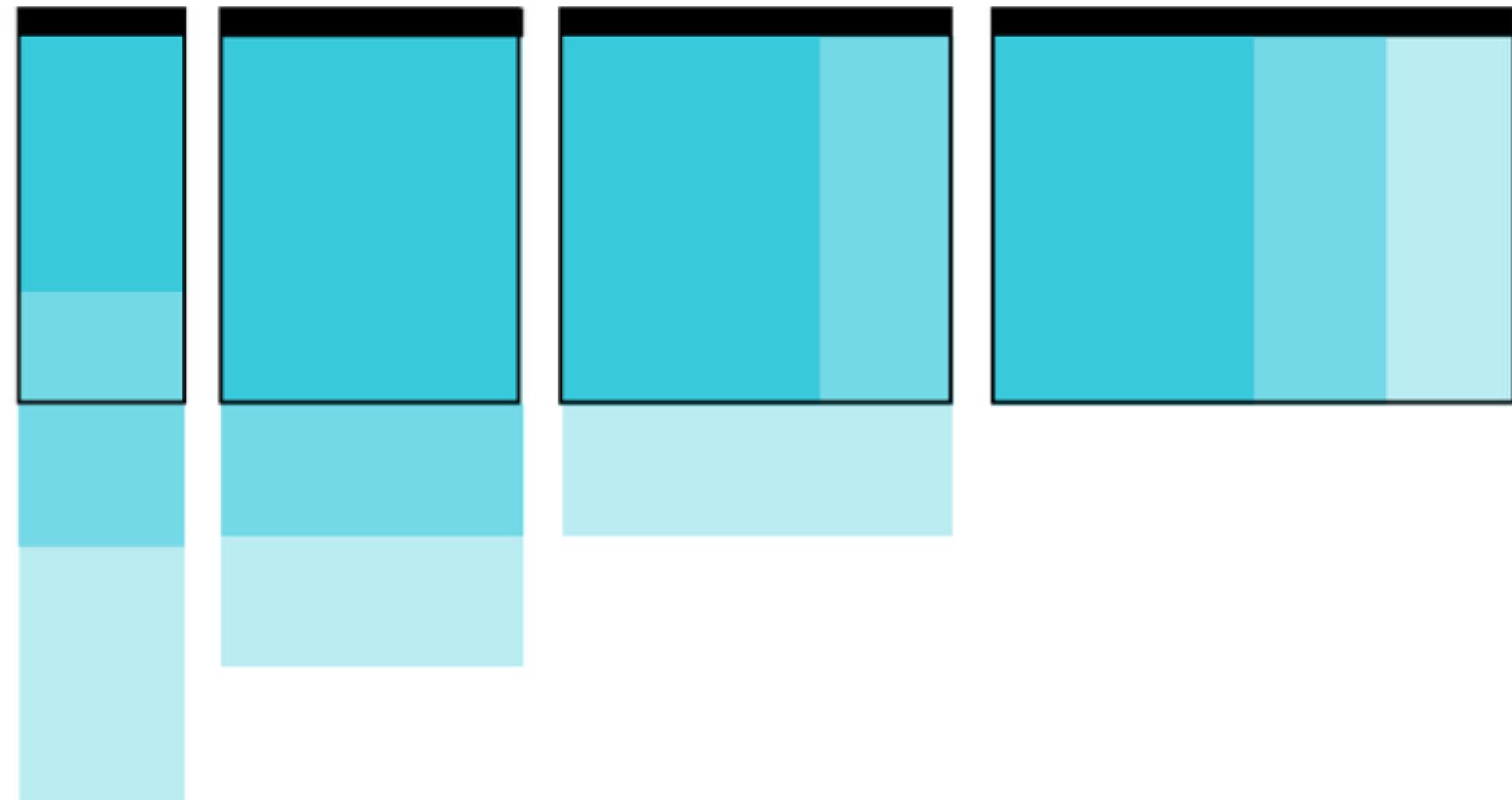
The MacBook screen shows a website with a dark header bar containing 'VIDEOS', 'BLOG', and 'ABOUT' buttons. To the left of these buttons is a small black box labeled 'Five Simple Steps'. The main content area has a light blue background with the text 'Talks from the best conferences' and a subtitle 'Watch the greatest in our industry give talks from the best conferences around the world.' Below this text are four video thumbnail images: a man in a green shirt speaking at a podium, a woman in a black top speaking on stage, a man standing on stage with a 'CYBERDILDONICS' slide, and a woman sitting at a desk with a laptop with 'UPFRONTCONF.COM' visible in the background.

<https://www.fivesimplesteps.com/pages/videos>



## Column drop

유연한 멀티 컬럼 그리드 레이아웃에 기초하여 화면 크기에 따라 콘텐츠를 단계적으로 떨궈 수직 방향으로 쌓아가게 하는 패턴이다.



기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



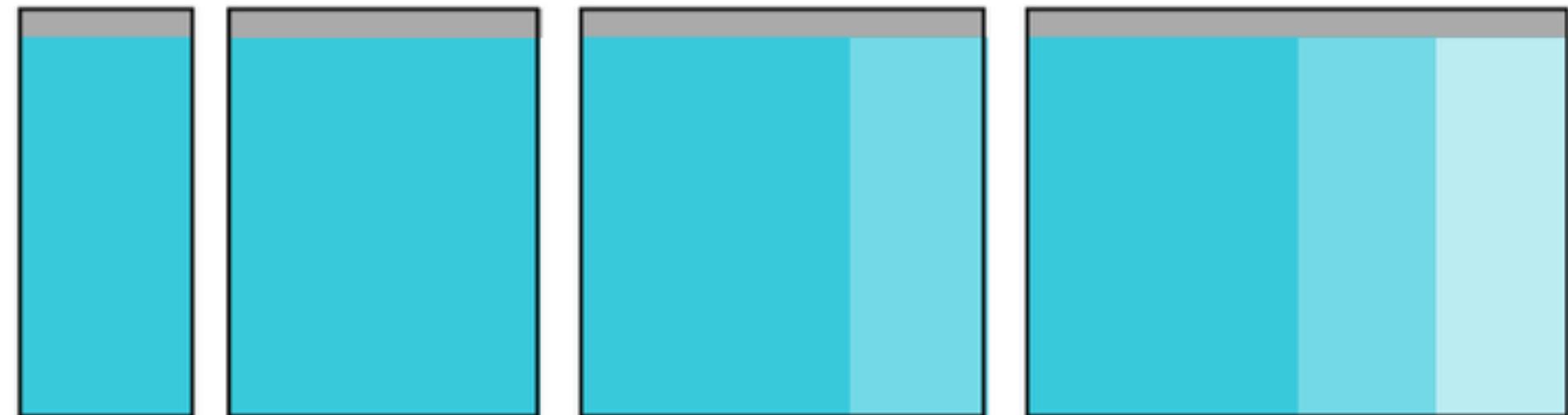
The image shows a responsive web design for a website titled "Introducing THE NOVEL". The design is centered on a MacBook screen, which is resting on a wooden stand. The website has a clean, modern layout with a white background. At the top, there is a large, bold title "THE NOVEL". Below the title, there is a detailed article about the definition and history of the novel. To the right of the text, there is a sidebar with three categories: "BILDUNGSROMAN", "THE CRIME NOVEL", and "THE HISTORICAL NOVEL", each with a small icon and a brief description. At the bottom of the website, the word "MacBook" is visible, indicating the device the site is being viewed on.

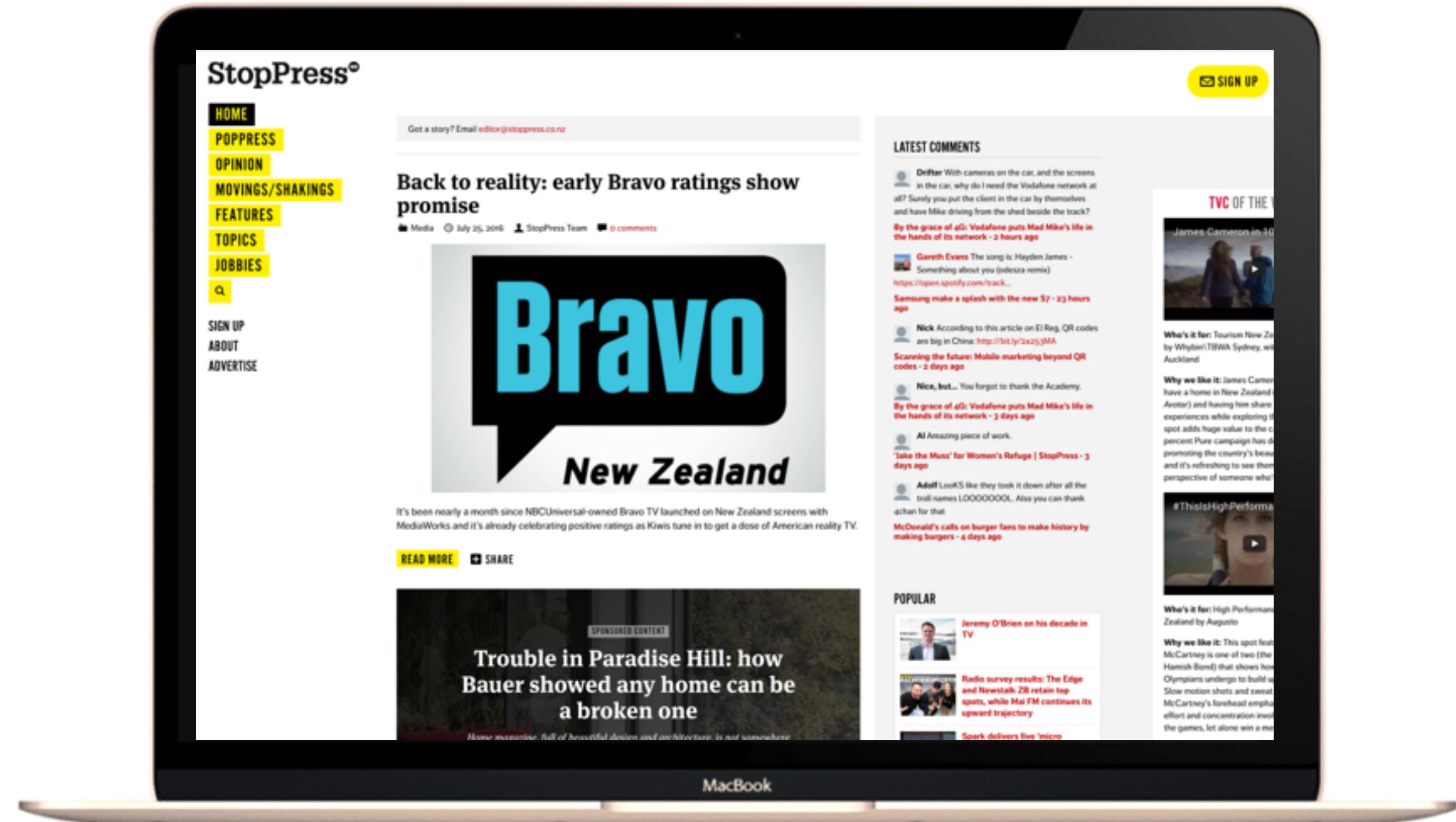
<http://introducingthenovel.com/>



## Column drop 2

Column drop 1과 비슷하지만, 화면 크기에 따라 콘텐츠를 단계적으로 감춘다는 점에서 다소 다른 패턴이다.

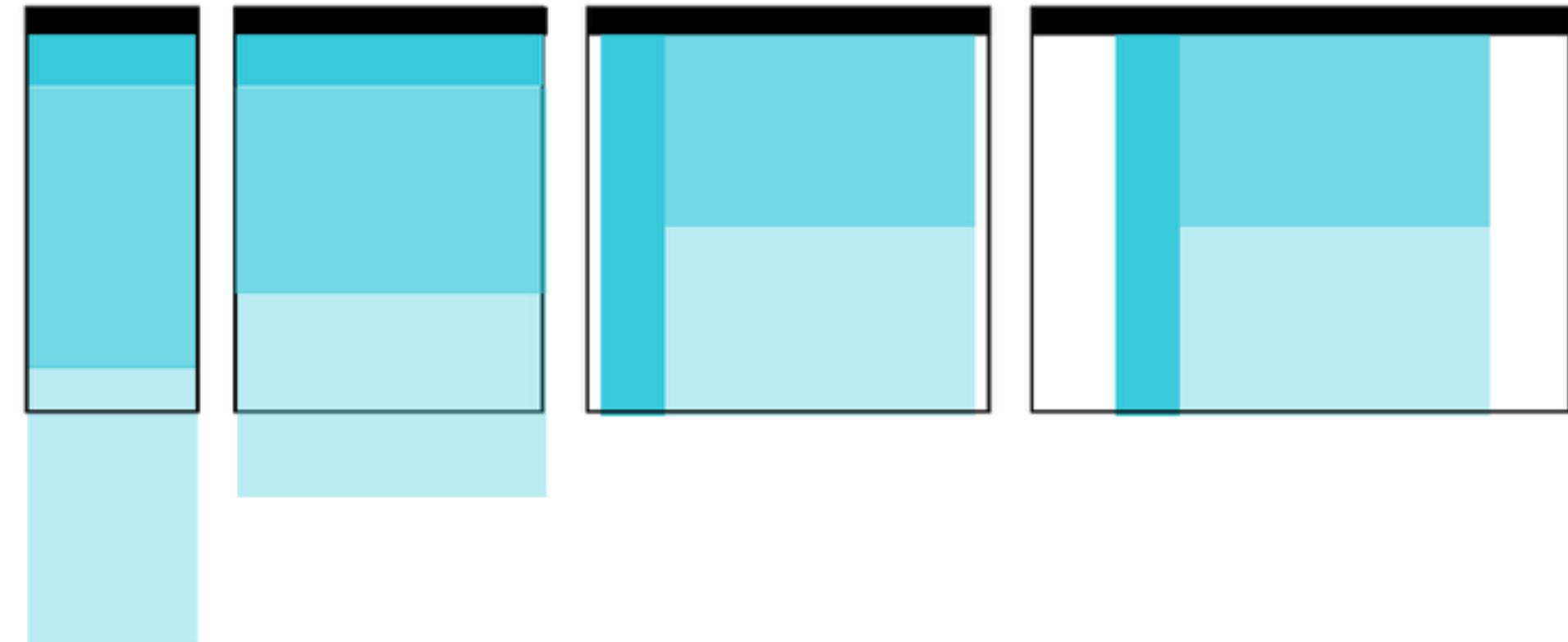




<http://stoppress.co.nz/>

## Layout Shifter

이 방법은 왼쪽에 위치한 내비게이션을 화면 크기에 따라 위쪽으로 이동시켜 좁은 공간에서 보다 효율적으로 탐색하기 위해 사용되는 패턴이다.

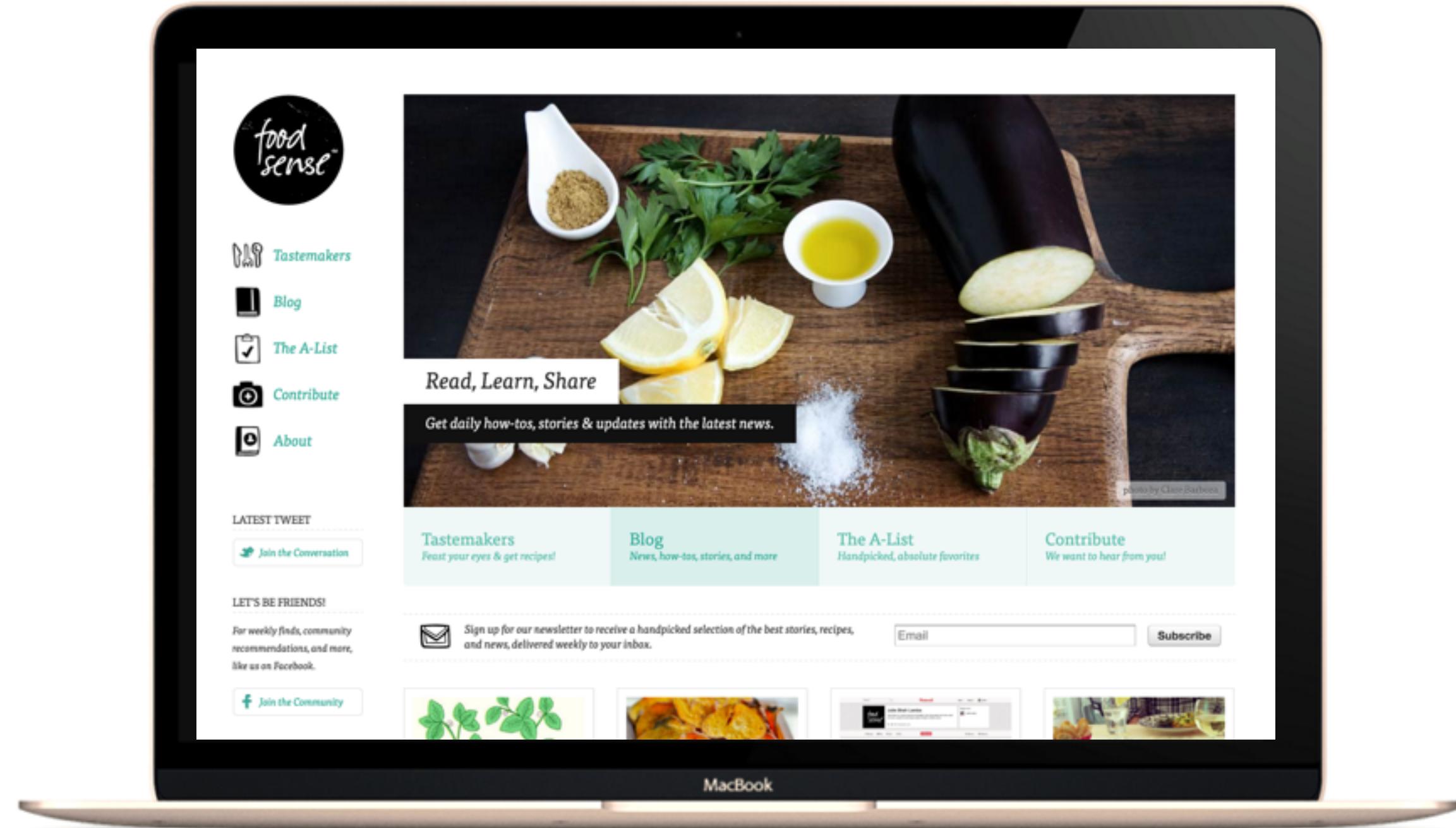


기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



기술을 이용해, 동료를 이용해, 세상을 이용해

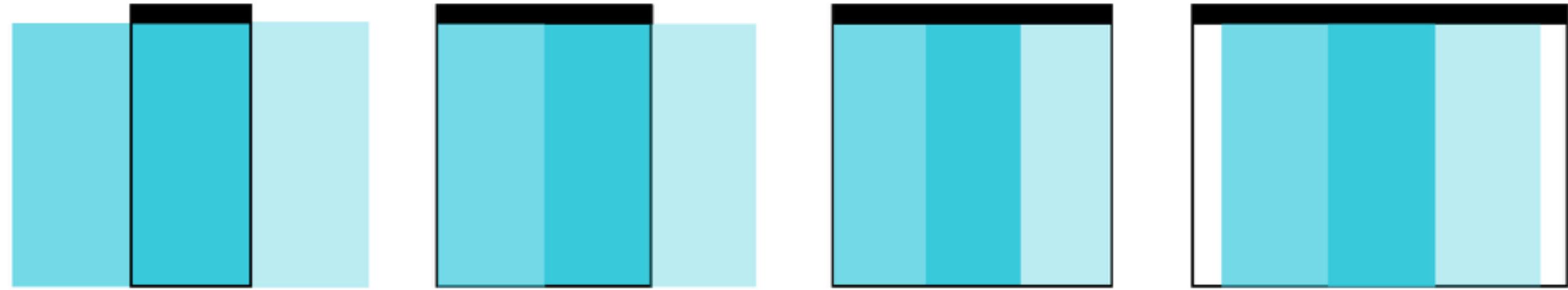


<http://foodsense.is/>



## Off Canvas

오프캔버스 방법은 좁은 공간을 효율적으로 사용할 수 있도록 감춰진 영역을 사용자의 요구에 따라 노출하는 패턴이다.



기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



기술을 이용해, 동료를 이용해, 세상을 이용해

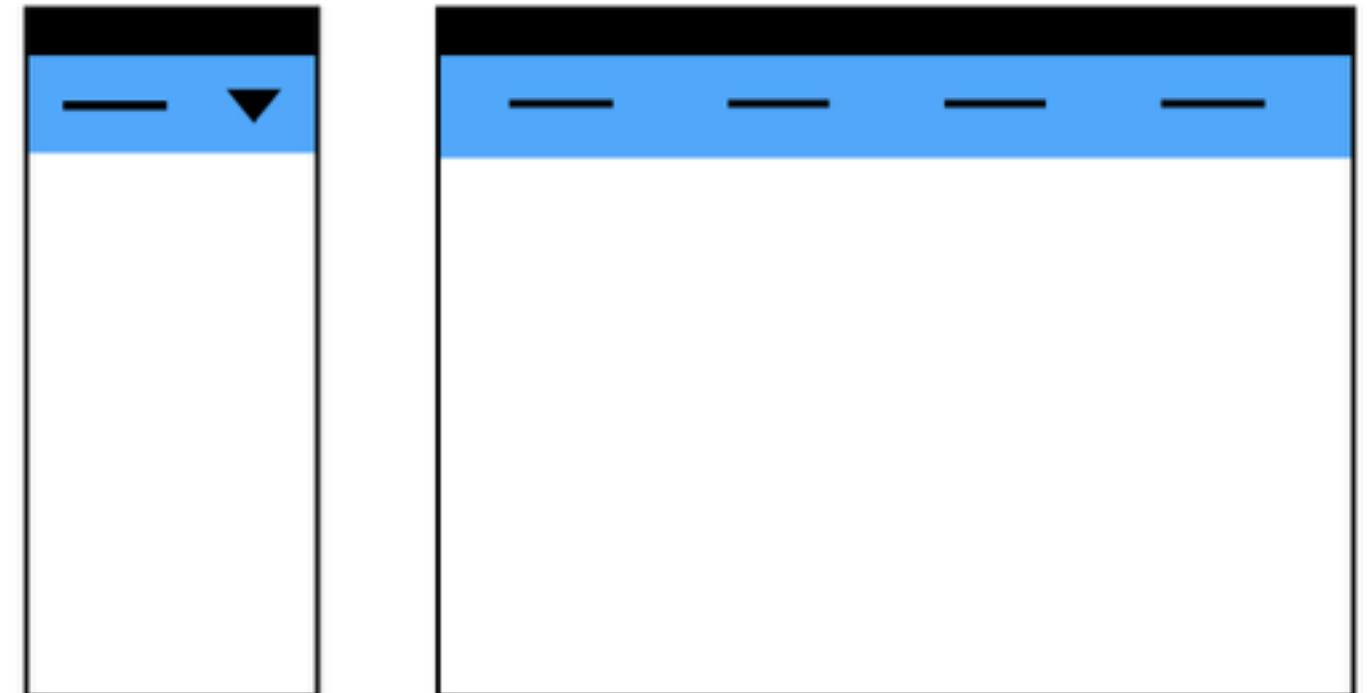


<https://juliemaurel.fr/>



## Select Navigation

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 셀렉트 메뉴로  
변경되는 패턴이다. (이전에 권장되던 방식)

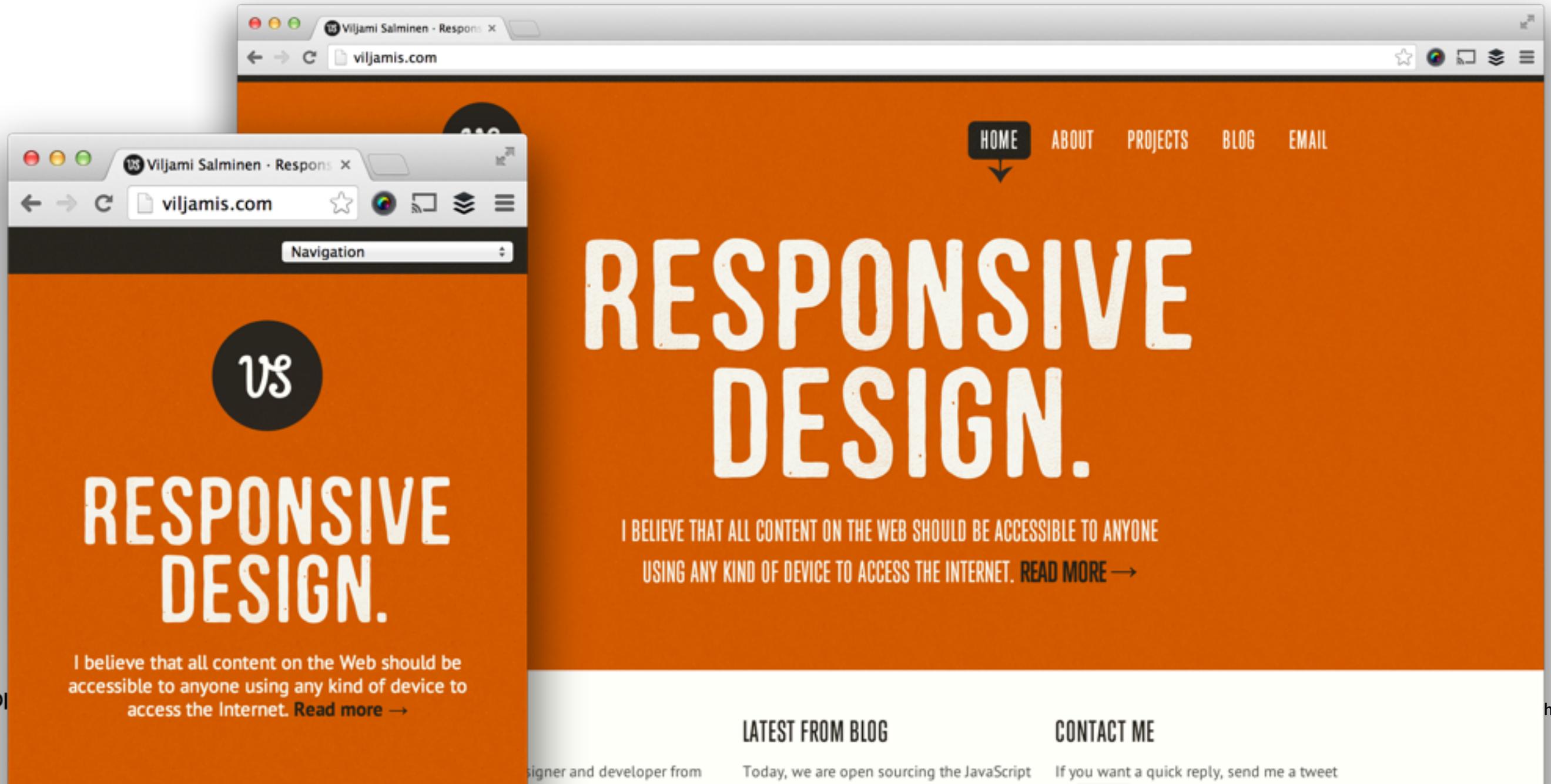


## Select Navigation

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 셀렉트 메뉴로  
변경되는 패턴 (이전에 권장되던 방식)

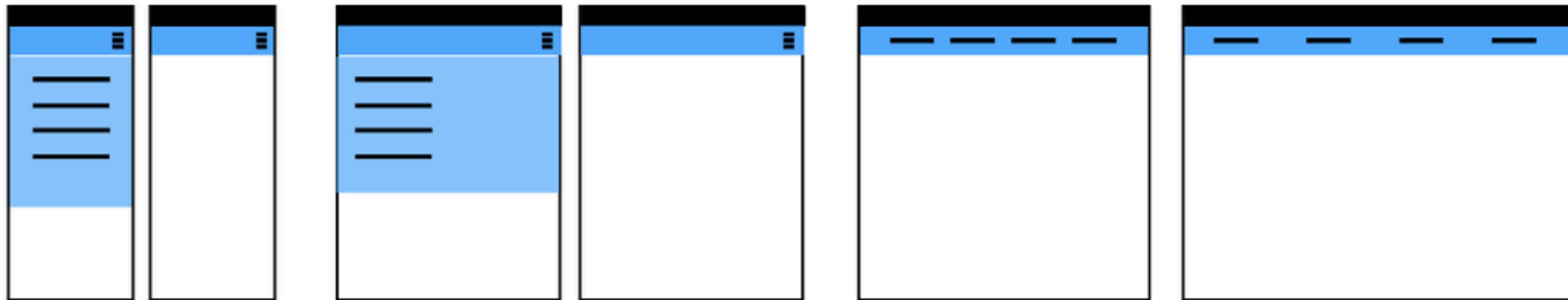
기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



## Toggle navigation

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 토글 버튼으로 바뀌어 사용자의 요구에 따라 펼쳐지는 패턴이다.



기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

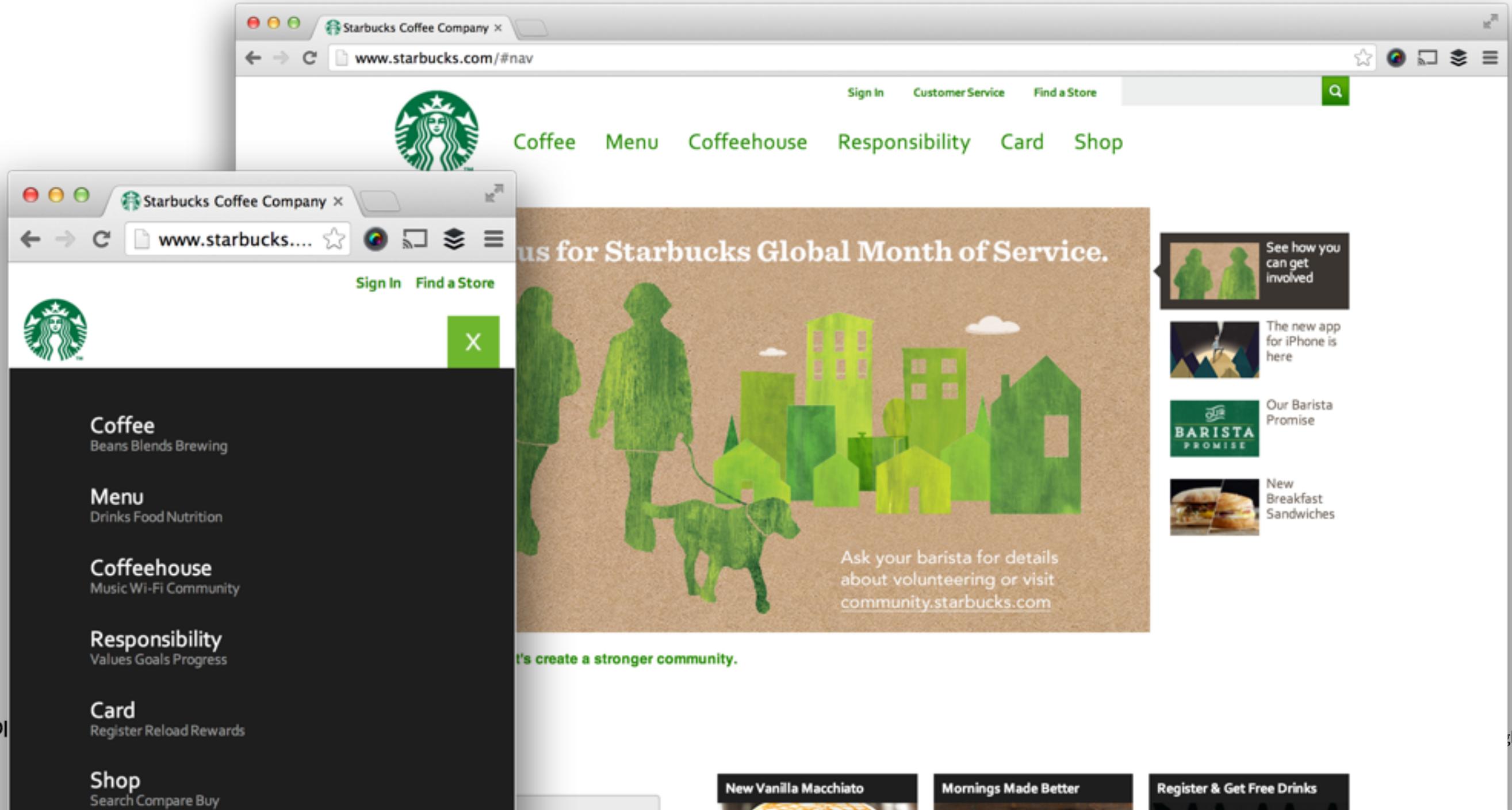


## Toggle navigation

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 토글 버튼으로 바뀌어 사용자의 요구에 따라 펼쳐지는 패턴이다.

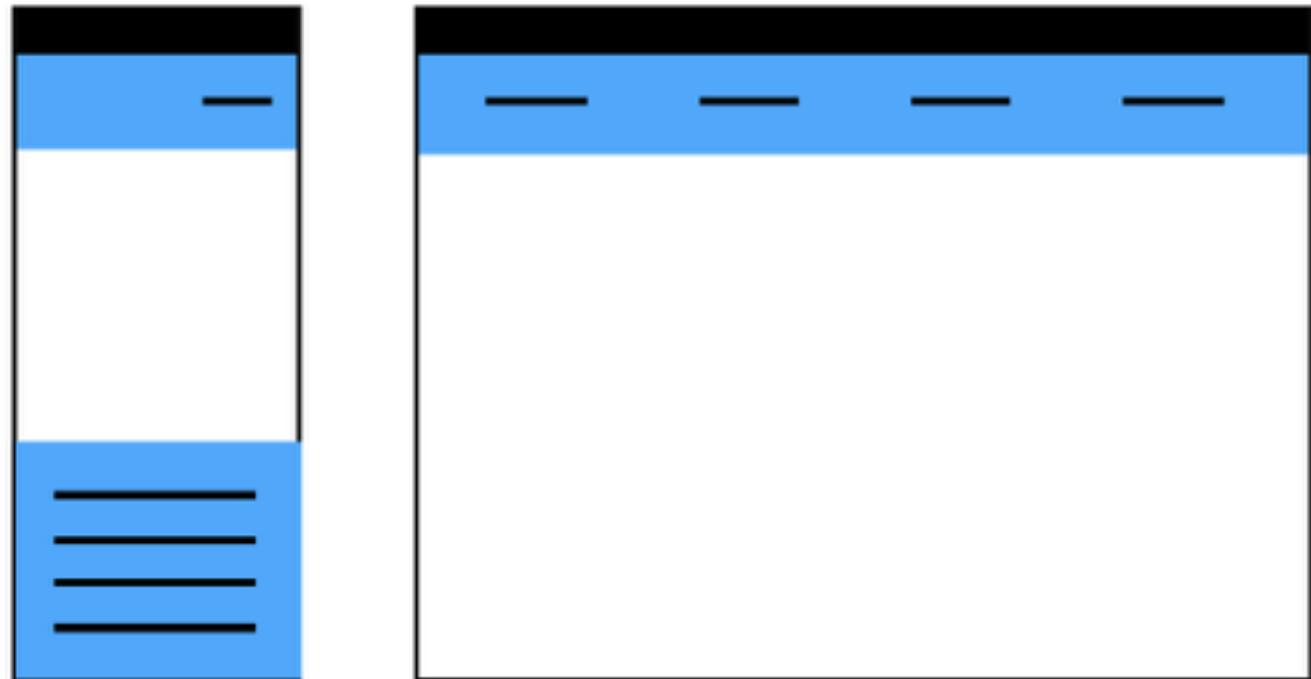
기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



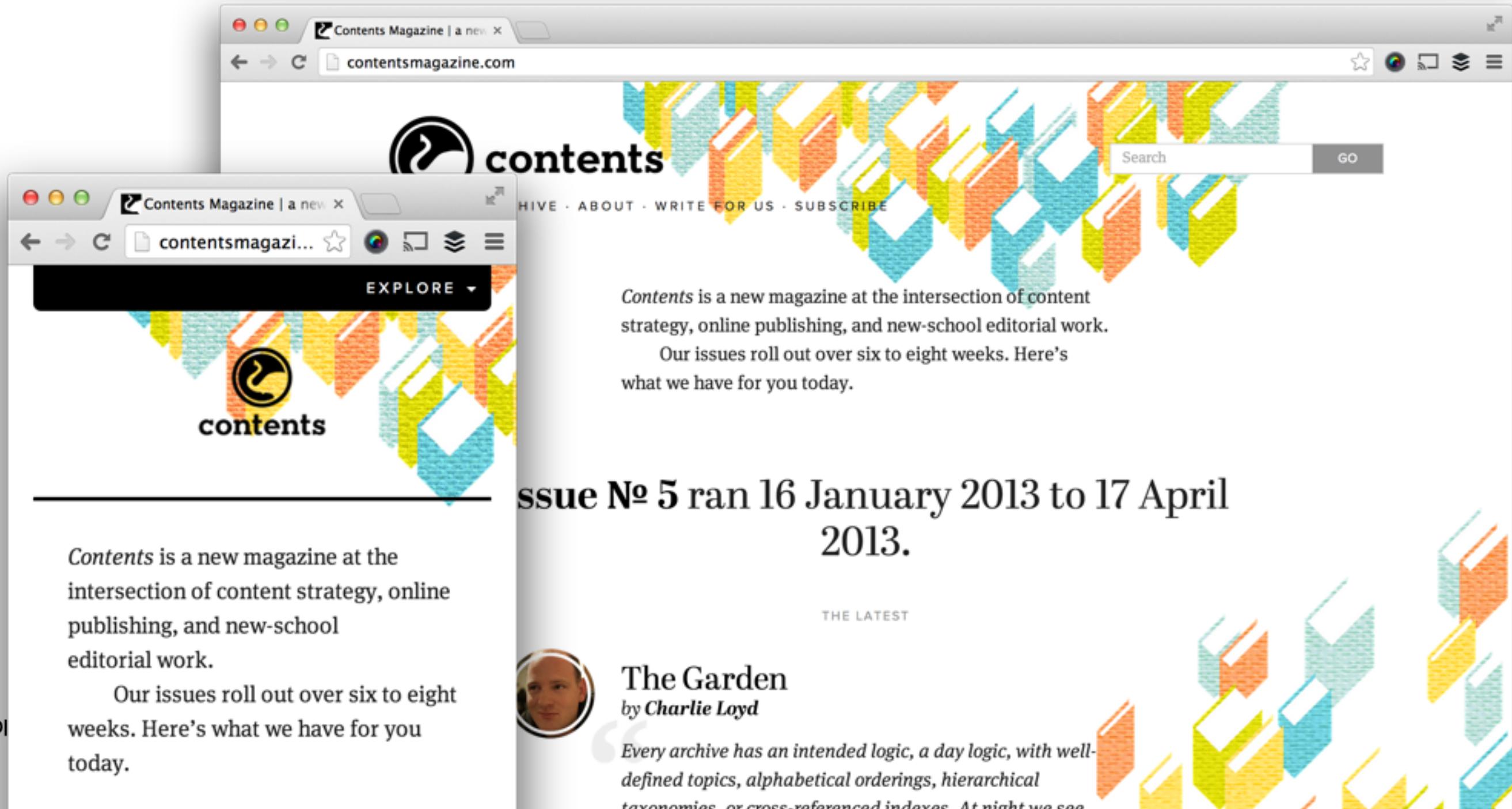
## Footer Navigation

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 하단 푸터 메뉴로 이동시키는 패턴이다.



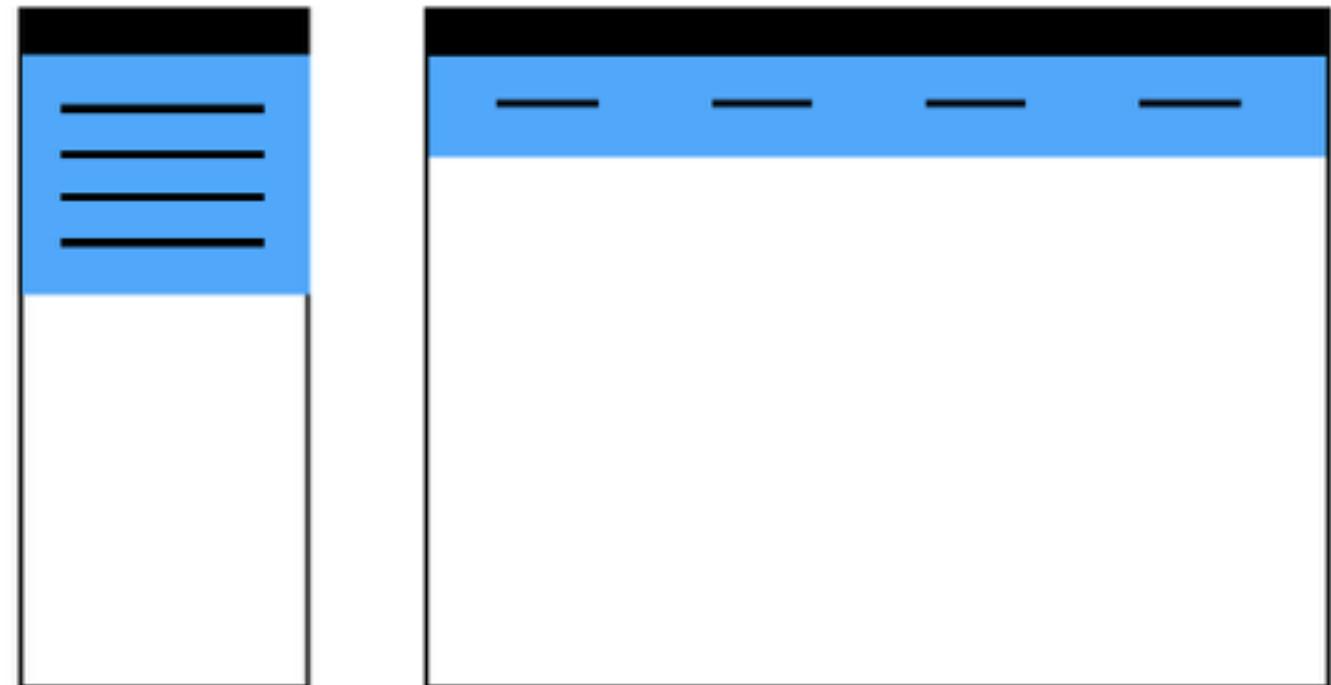
## Footer Navigation

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 하단 푸터 메뉴로 이동시키는 패턴이다.



## Stacked top links navigation

글로벌 내비게이션 바 영역의 메뉴들을 모바일 화면에서 항상 화면 위에 노출시키는 패턴이다.



## Stacked top links navigation

글로벌 내비게이션 바 영역의 메뉴들을 모바일 화면에서 항상 화면 위에 노출시키는 패턴이다.

The screenshot illustrates the 'Stacked Top Links' navigation pattern. In the top browser window, a horizontal navigation bar is visible with links: Home, About, Products, Services, and Contact. In the bottom browser window, the same navigation items are shown stacked vertically on top of each other. The main content area contains the following text:

# Stacked Top Links

A navigation that vertically stacks navigation items on top of each other for small screens, and becomes a horizontal list when screen space becomes available.

## Pros

- Easy to implement-** you can implement your large-screen site almost as-is.
- More touch-friendly than the top links pattern** - Where links across the top can lead to unwanted proximity clicks, stacking the links provides a larger hit state

## Cons

- Height issues-** Height matters in mobile. As Luke's book explains, content-first, nav-second is preferred for mobile web experiences. You want to get the users to the meat-and-potatoes content as quickly as possible. That means getting the navigation out of user's way so they can focus on the core information on the page. It can also be confusing when all the core content is cut off

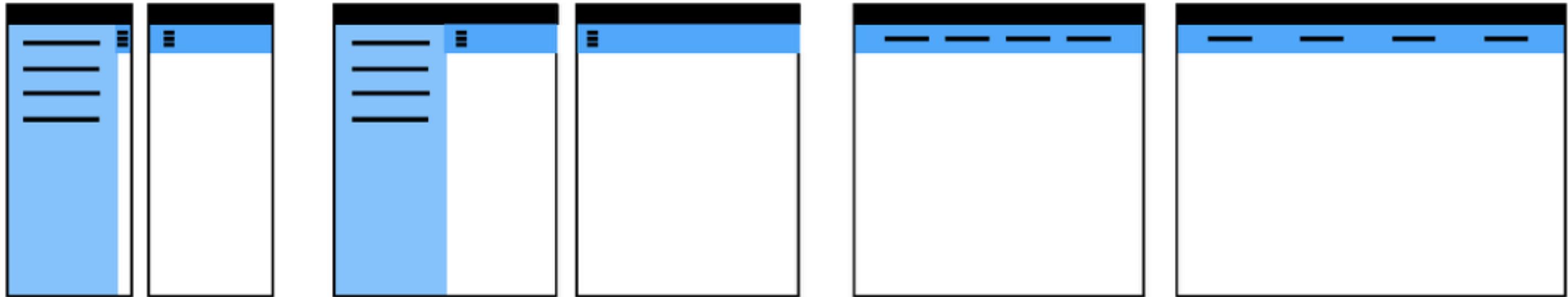
02

# Stacked Top Links



## Left Nav Flyout

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 좌측으로 감춰진 상태에서 사용자의 요구에 따른 인터랙션으로 메뉴가 노출되는 패턴이다.



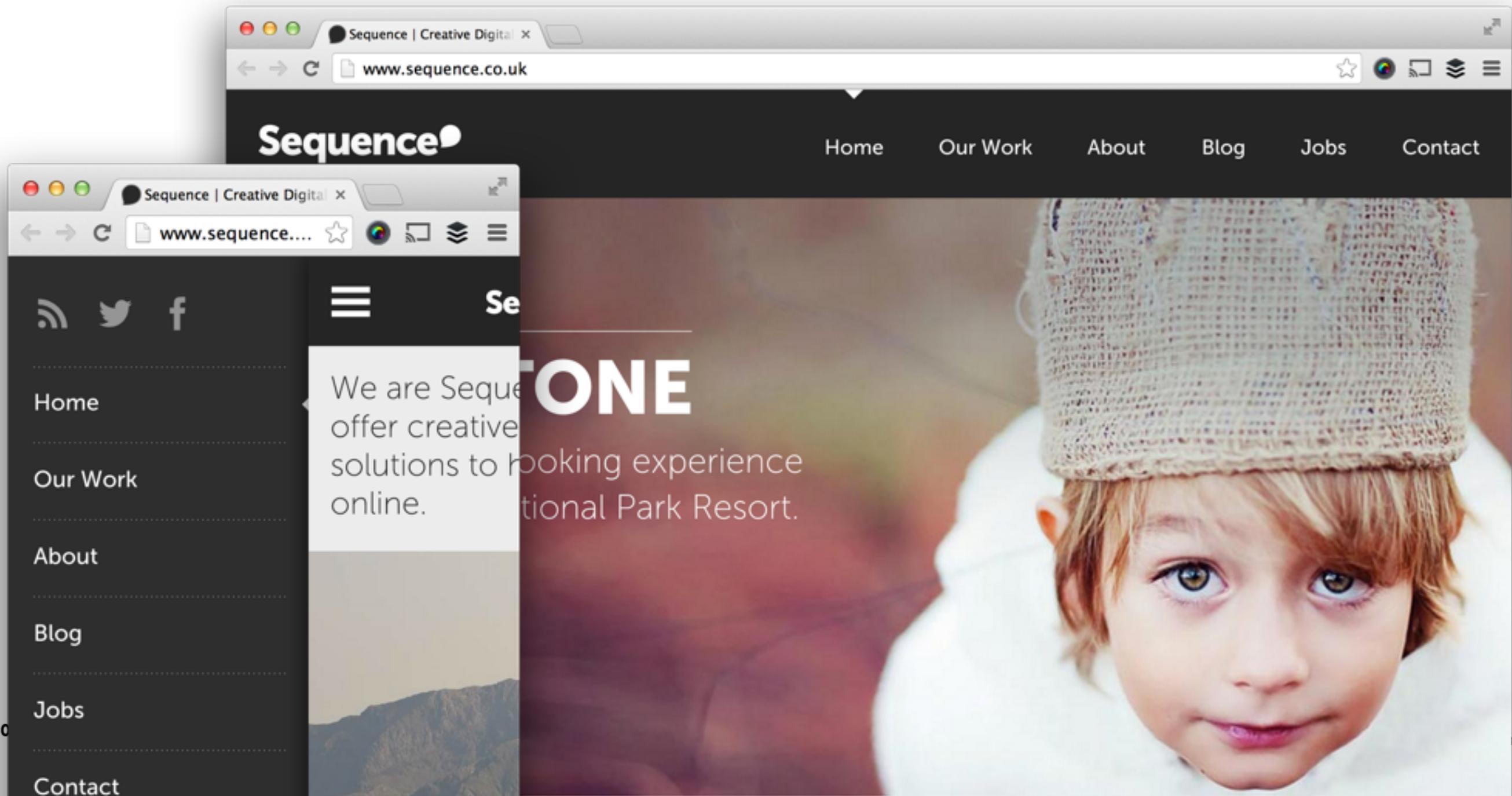
기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



## Left Nav Flyout

글로벌 내비게이션 바 영역의 메뉴들이 모바일 화면에서 좌측으로 감춰진 상태에서 사용자의 요구에 따른 인터랙션으로 메뉴가 노출되는 패턴이다.



## Priority + Navigation

글로벌 내비게이션 바 영역의 메뉴 중 일부를 모바일 환경 화면에서 감추는 패턴이다.



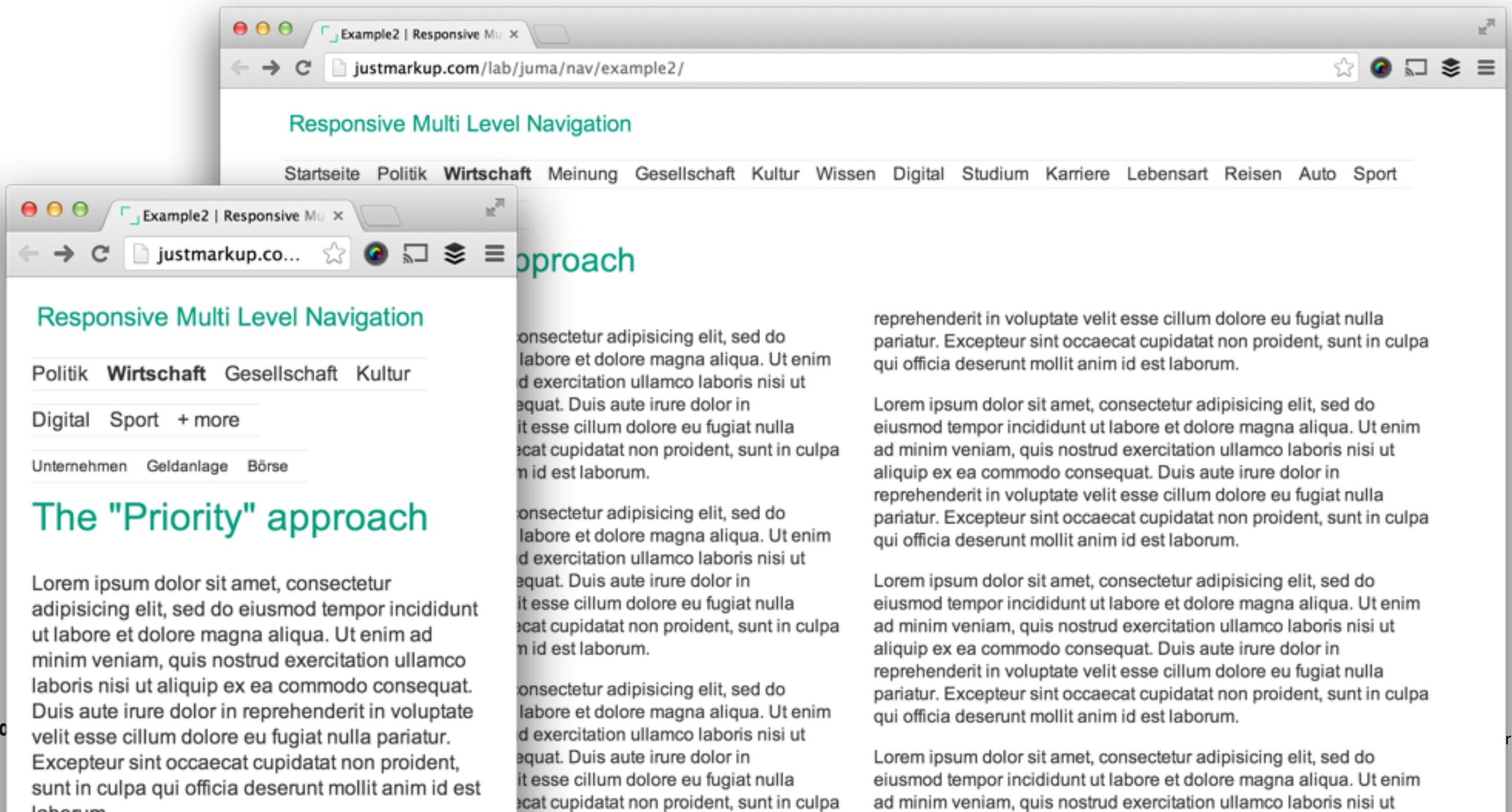
기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



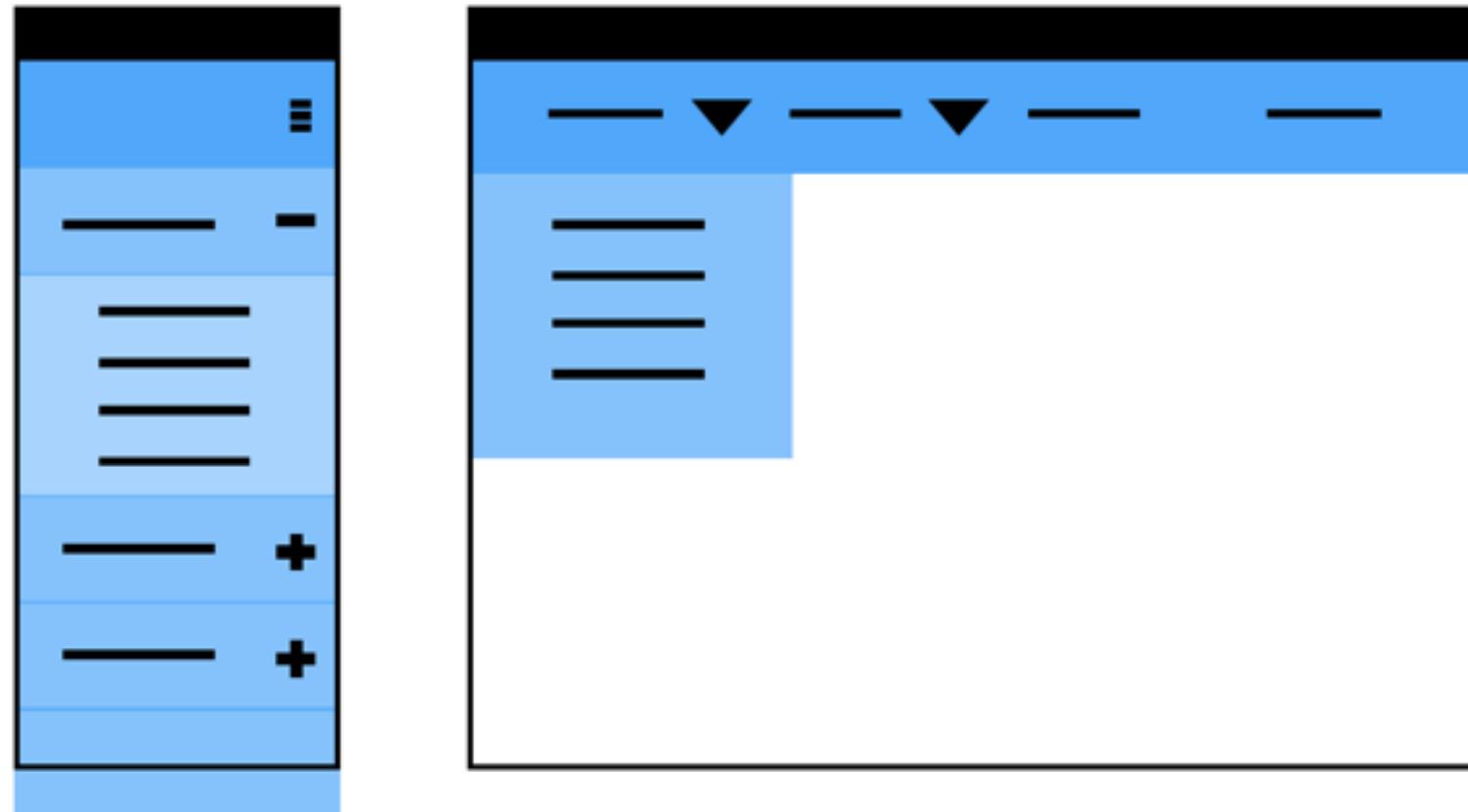
# Priority + Navigation

글로벌 내비게이션 바 영역의 메뉴 중 일부를 모바일 환경 화면에서  
감추는 패턴이다.



## Priority + Navigation 2

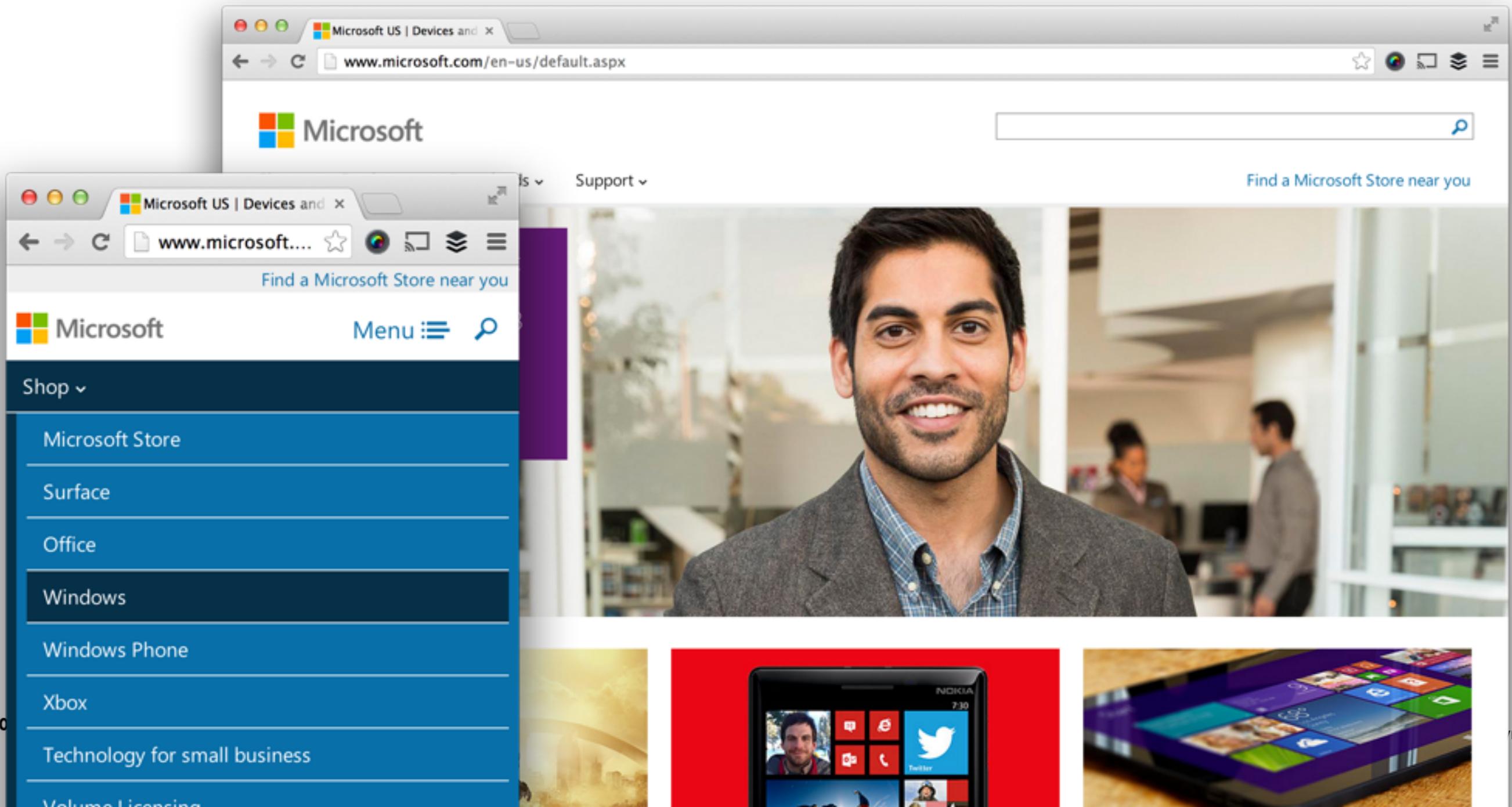
하위 메뉴를 포함하는 글로벌 내비게이션 바 영역의 메뉴를 모바일 환경 화면에서 펼쳐 열어볼 수 있는 패턴이다.



## Priority + Navigation 2

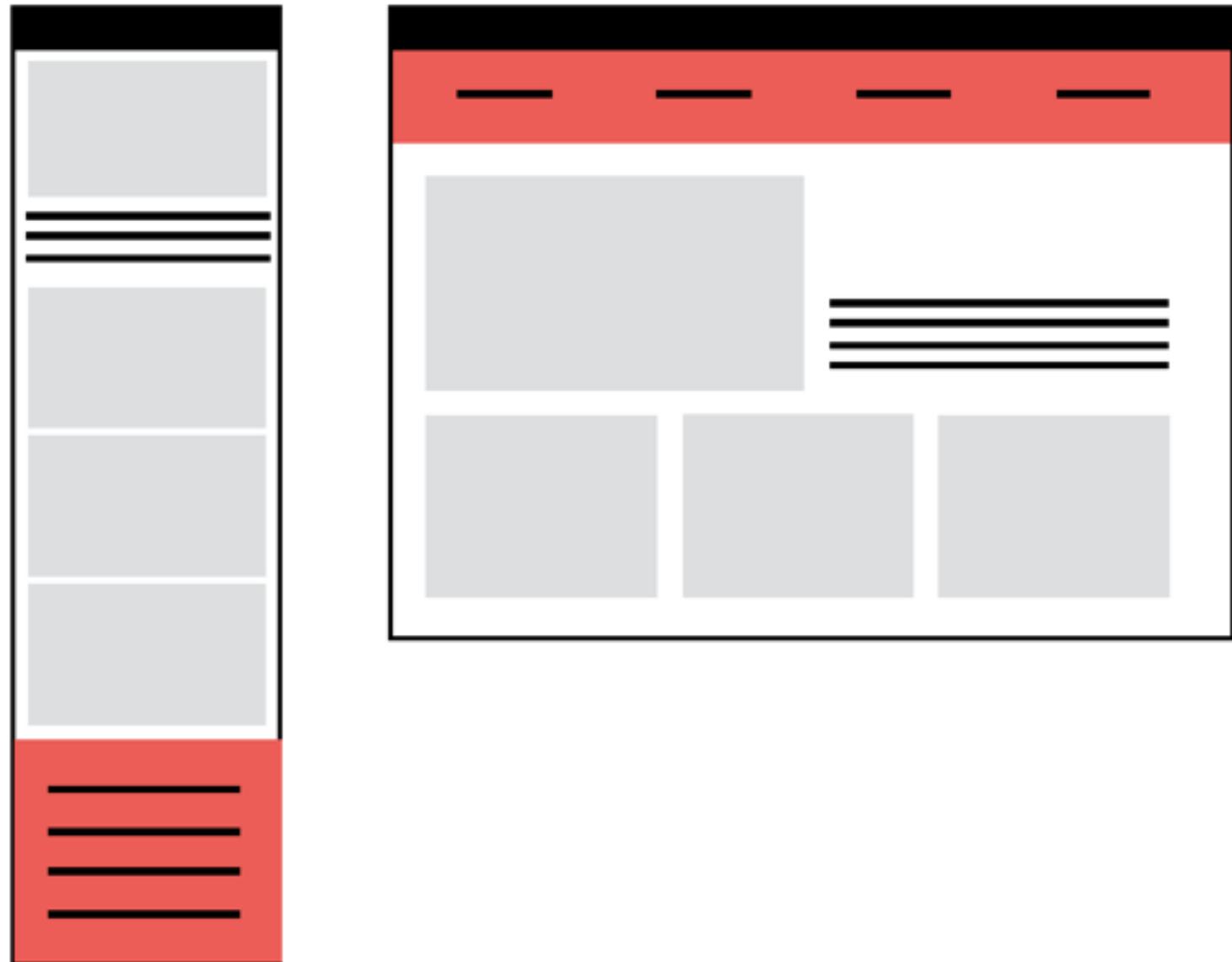
하위 메뉴를 포함하는 글로벌 내비게이션 바 영역의 메뉴를 모바일 환경 화면에서 펼쳐 열어볼 수 있는 패턴이다.

기술을 이용해, 동료를 이용해, 세상을 이용해



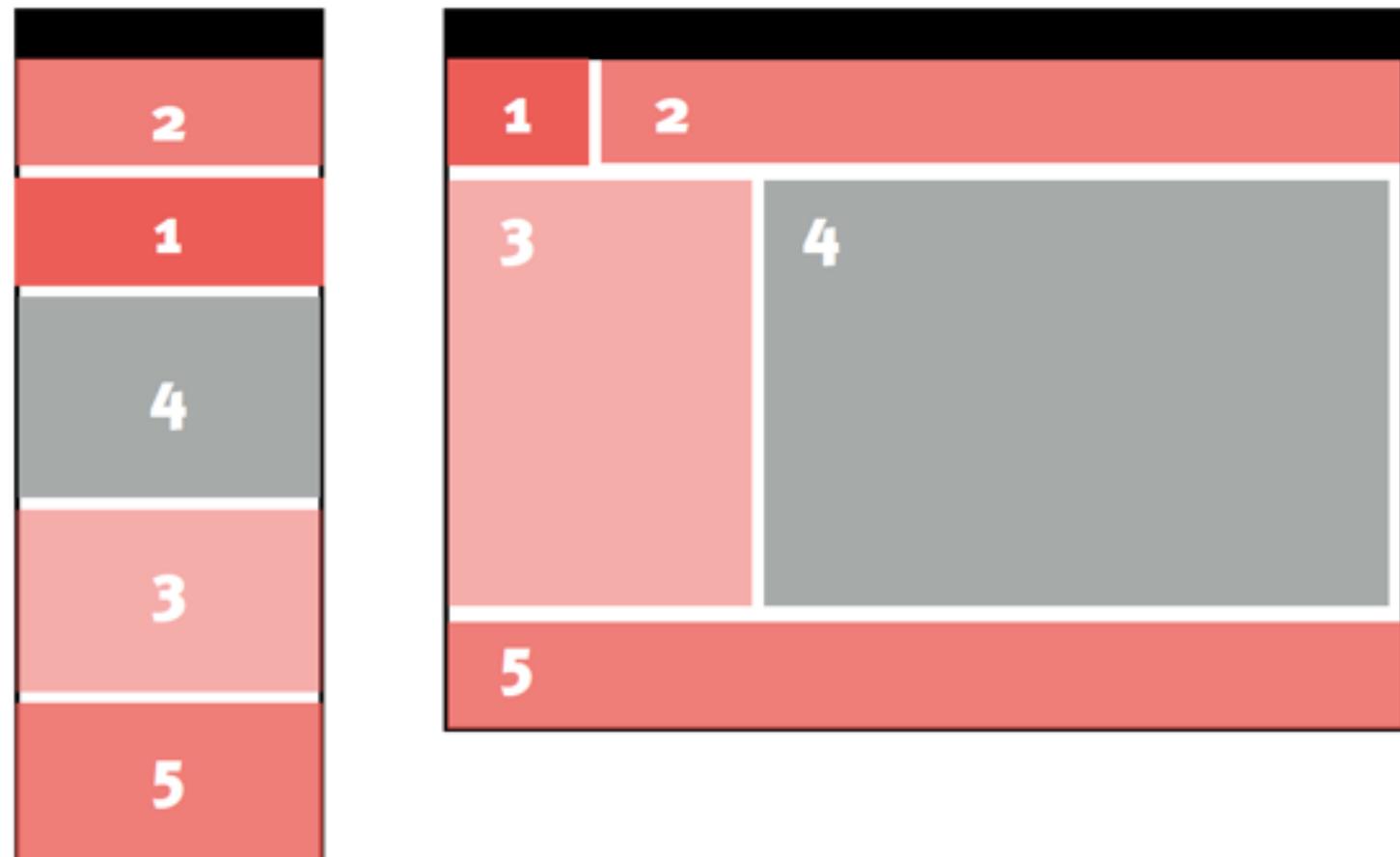
## Source order

콘텐츠에 따라 중단점(Breakpoint) 설계에 맞춰 순서(Order) 또는 위치(Position)가 변경 된다.



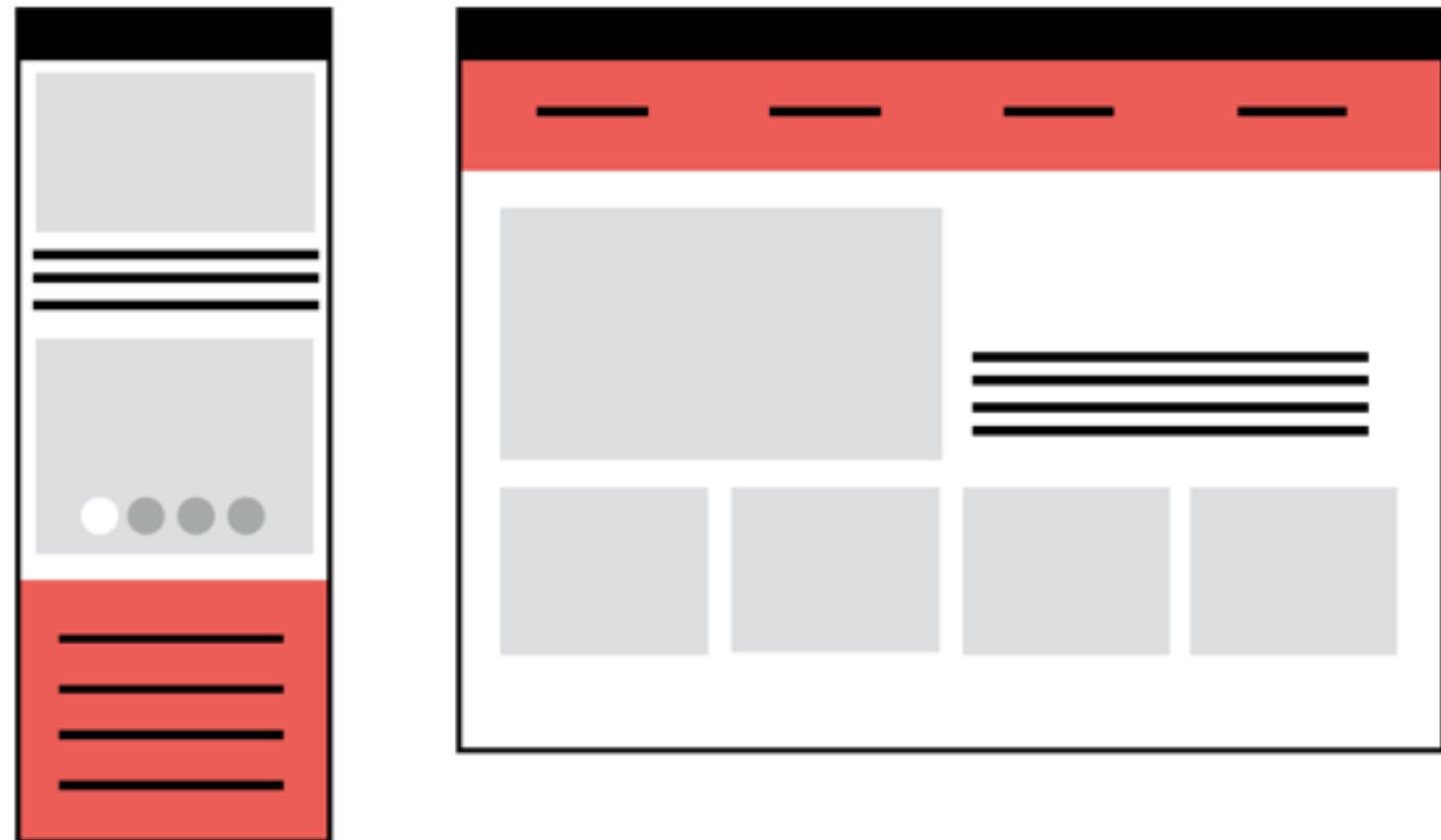
## Order shift

RWD 콘텐츠 구성은 유연해야 하므로 중단점 설계에 따라 콘텐츠 순서를 바꾸기 용이하도록 제작한다.



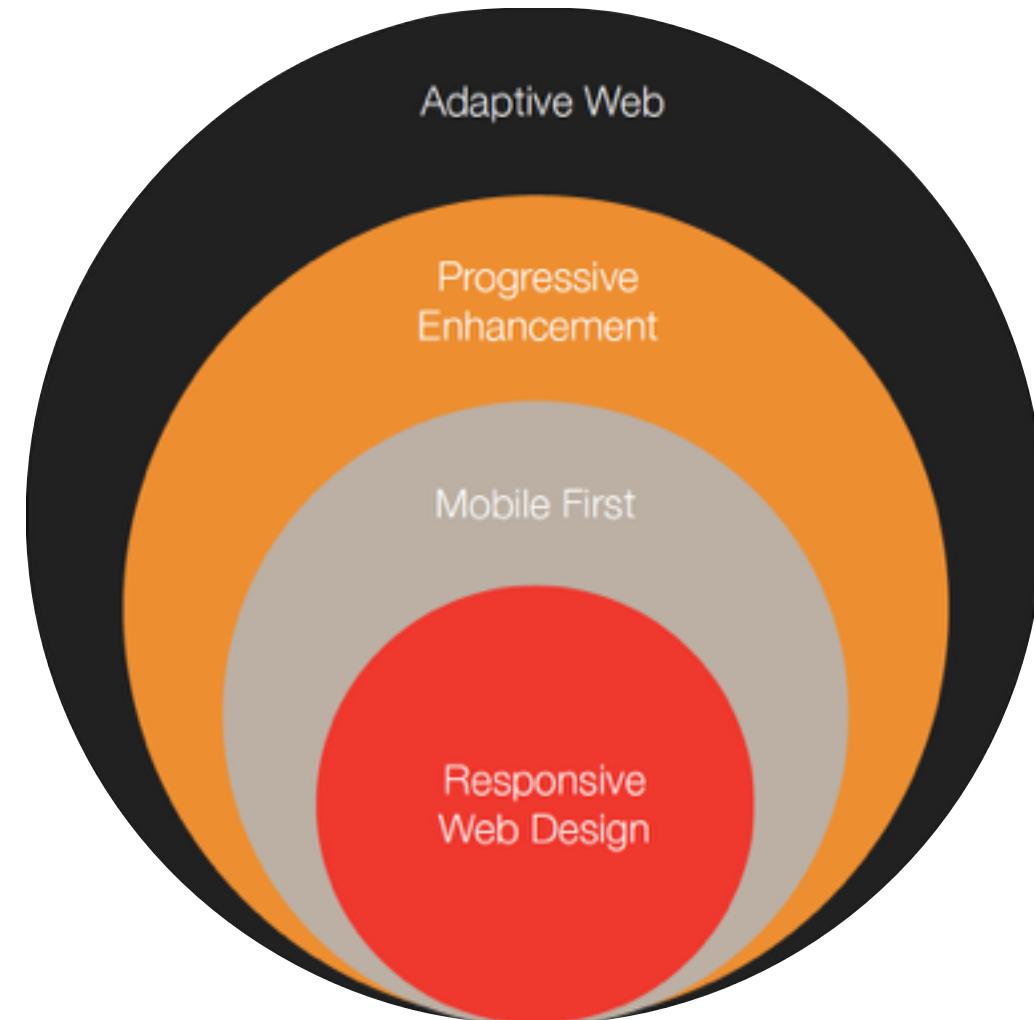
## Grouping contents

중단점 설계에 따라 많은 양의 콘텐츠를 작은 공간에서 효율적으로 보여줄 수 있도록 콘텐츠를 모아 캐로셀처럼 제공할 수 있다. (단, 접근성을 고려해야 함)



# 반응형 웹 디자인 vs 적응형 웹 무엇이 다른 것인가?

기술을 이용해, 동료를 이용해, 세상을 이용해



## Adaptive Web

「하나의 웹(One Web)」 실현을 위한 큰 그림

## Progressive Enhancement

기술적 접근성 마인드 설정

## Mobile First

모바일 우선 컨텍스트 마인드 설정

## Responsive Web Design

유연한 레이아웃/디자인 방법론

# 점진적 기능향상이란?

기술을 이용해, 동료를 이용해, 세상을 이용해



## 점진적 기능 향상(Progressive Enhancement)

웹 브라우저에 관한 이야기가 아닙니다. 기술적 제약에 상관없이 사용자가 콘텐츠에 대한 접근이 가능하도록 제공하는 것을 말하는 것입니다.

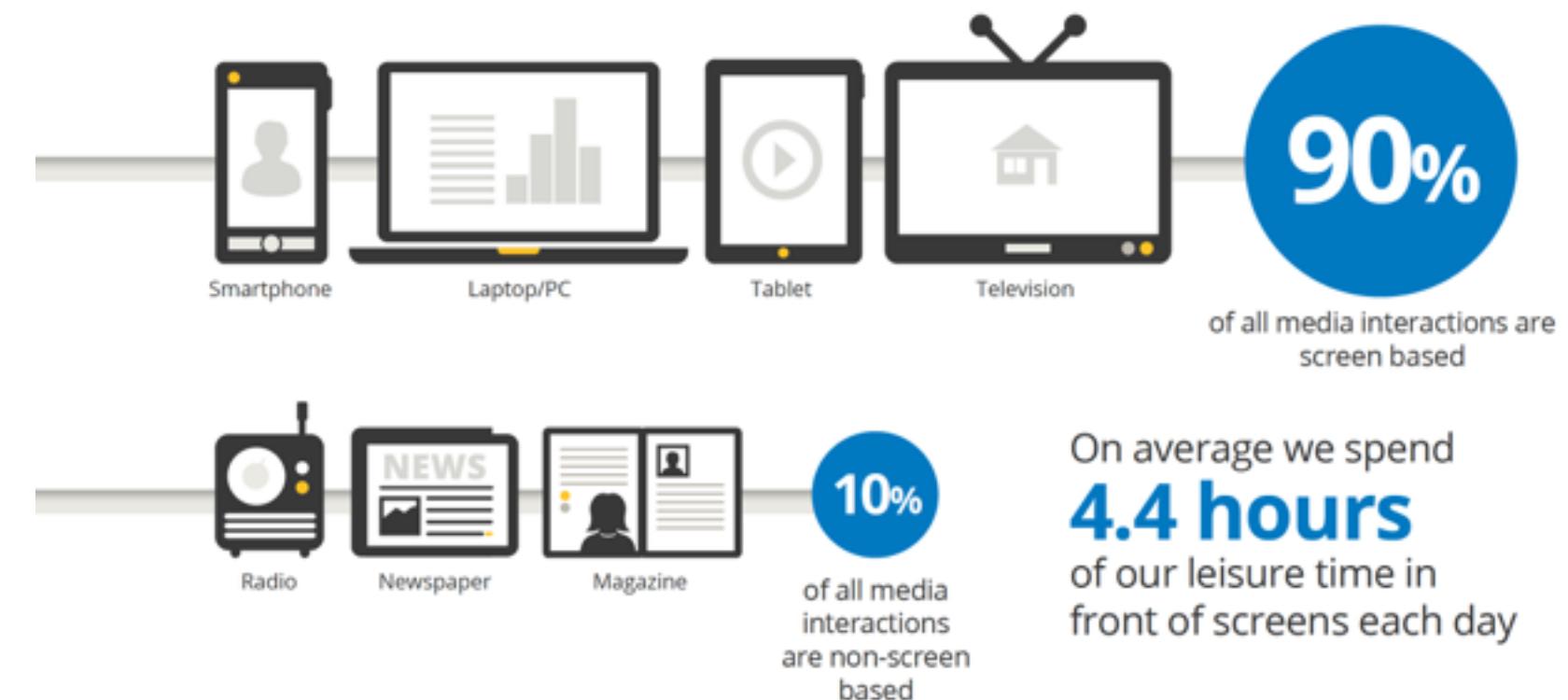
점진적 기능 향상 뿐만 아니라,  
우아한 기능 저하 방식도 고려하여  
기술에 관계없이 서비스를 제작한다.

아론 구스타프손(Aaron Gustafson), Adaptive Web Design

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

Majority of our daily media interactions are screen based



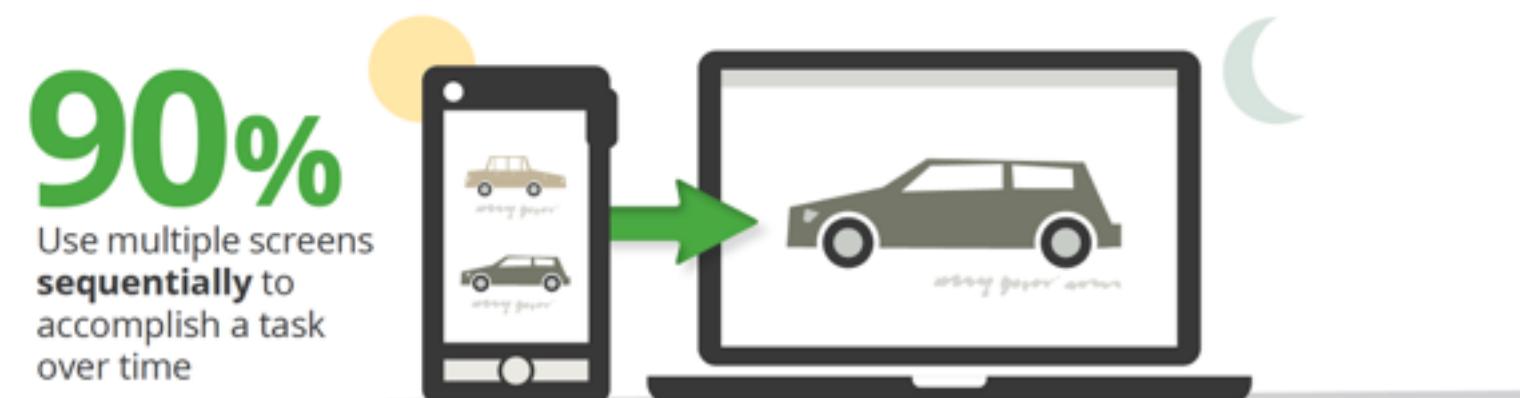
Google  
The New Multi-screen World:  
Understanding Cross-platform Consumer Behavior  
August 2012

사람들이 가장 많이 사용하는 매체는  
스크린이다.

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

**Sequential screening is common &  
mostly completed within a day**



**98% move between devices that same day**

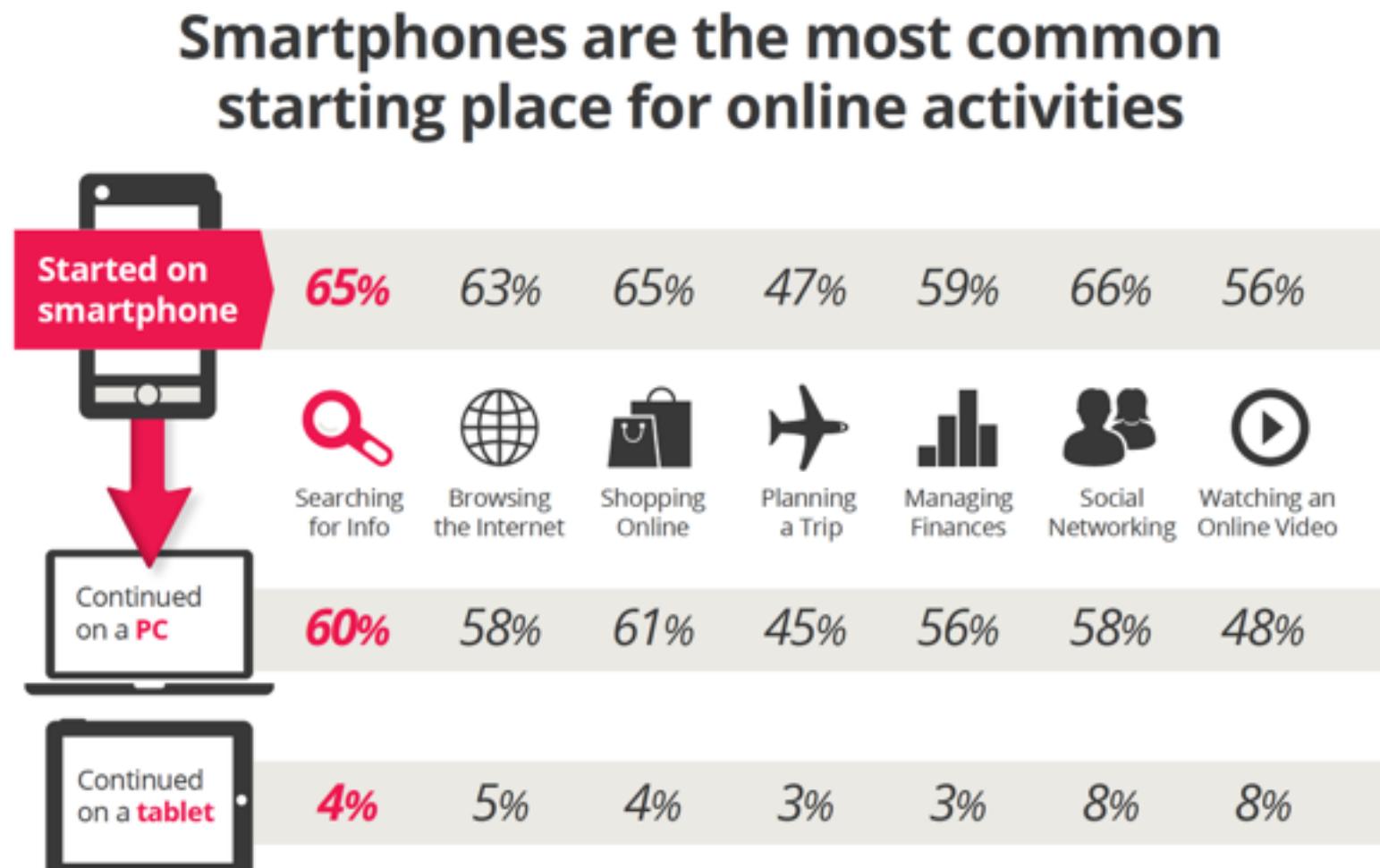


사람들은 하나의 기기가 아닌  
여러 기기 사이를 순환한다.

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



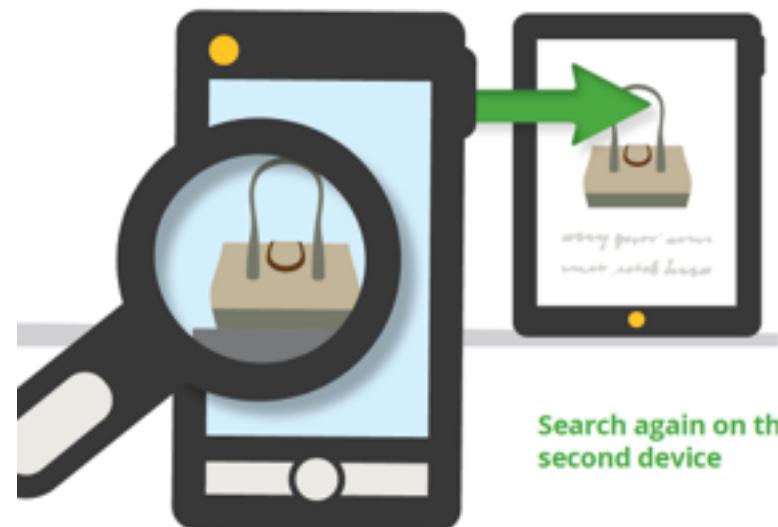
사람들은 휴대용 기기인  
스마트폰을 가장 많이 사용한다.

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

## Consumers rely on search to move between devices



	Searching for Info	Browsing the Internet	Shopping Online	Watching an Online Video
Search again on the second device	63%	61%	51%	43%
Directly navigating to the destination site	52%	58%	48%	43%
Via email / sending a link to myself	49%	45%	31%	30%

Google  
The New Multi-screen World:  
Understanding Cross-platform Consumer Behavior  
  
August 2012

사람들은 스마트폰을 이용해 검색한 후,  
다른 기기로 이동하는 경향을 보인다.

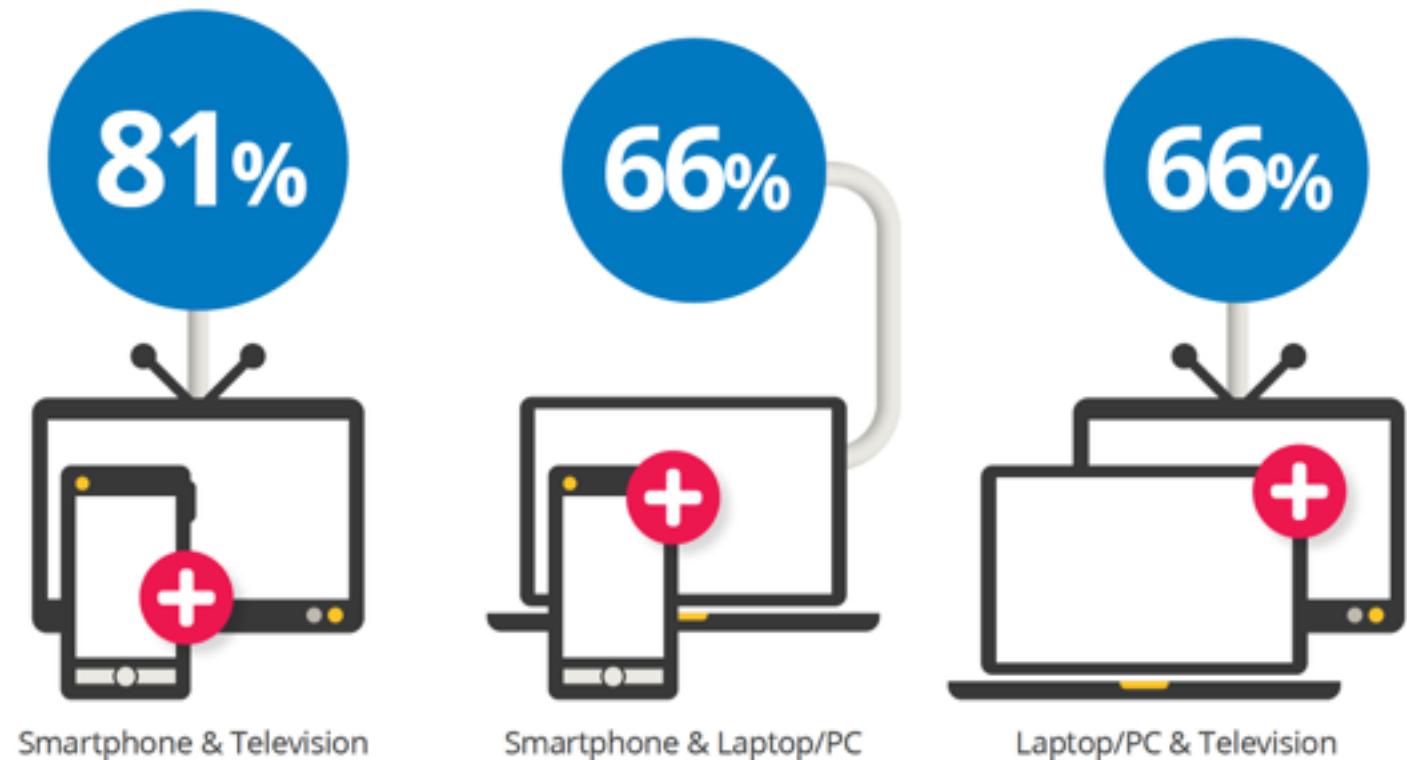
# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

We also multi-screen by using more than  
one device simultaneously

We use an average of three different screen combinations every day

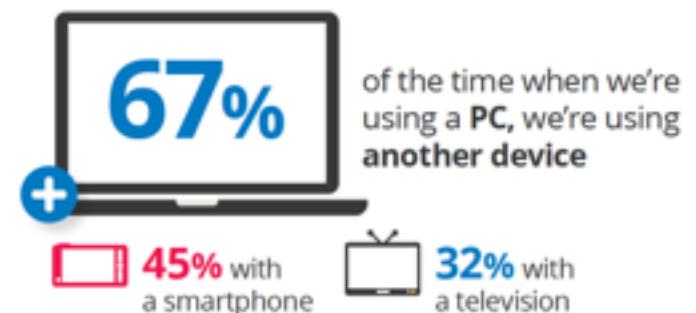
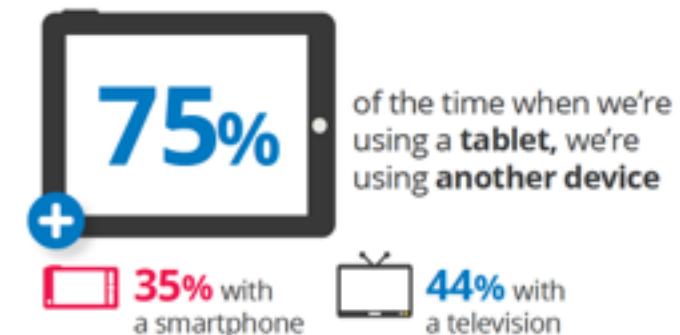
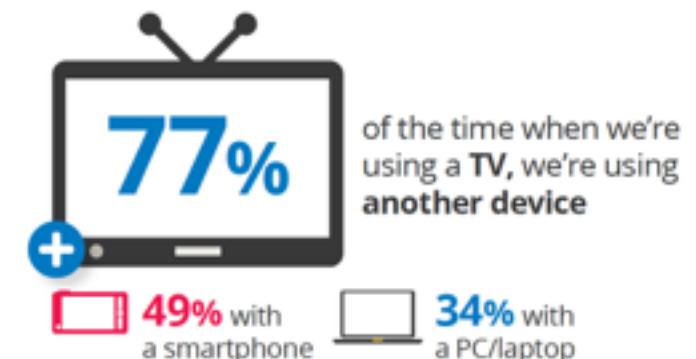
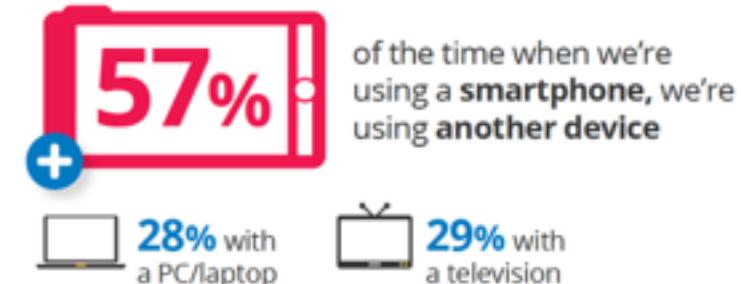


사람들은 스마트폰을 포함하여  
동시에 여러 기기를 사용한다.  
(멀티 스크린)

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

## Smartphones are the most frequent companion devices during simultaneous usage



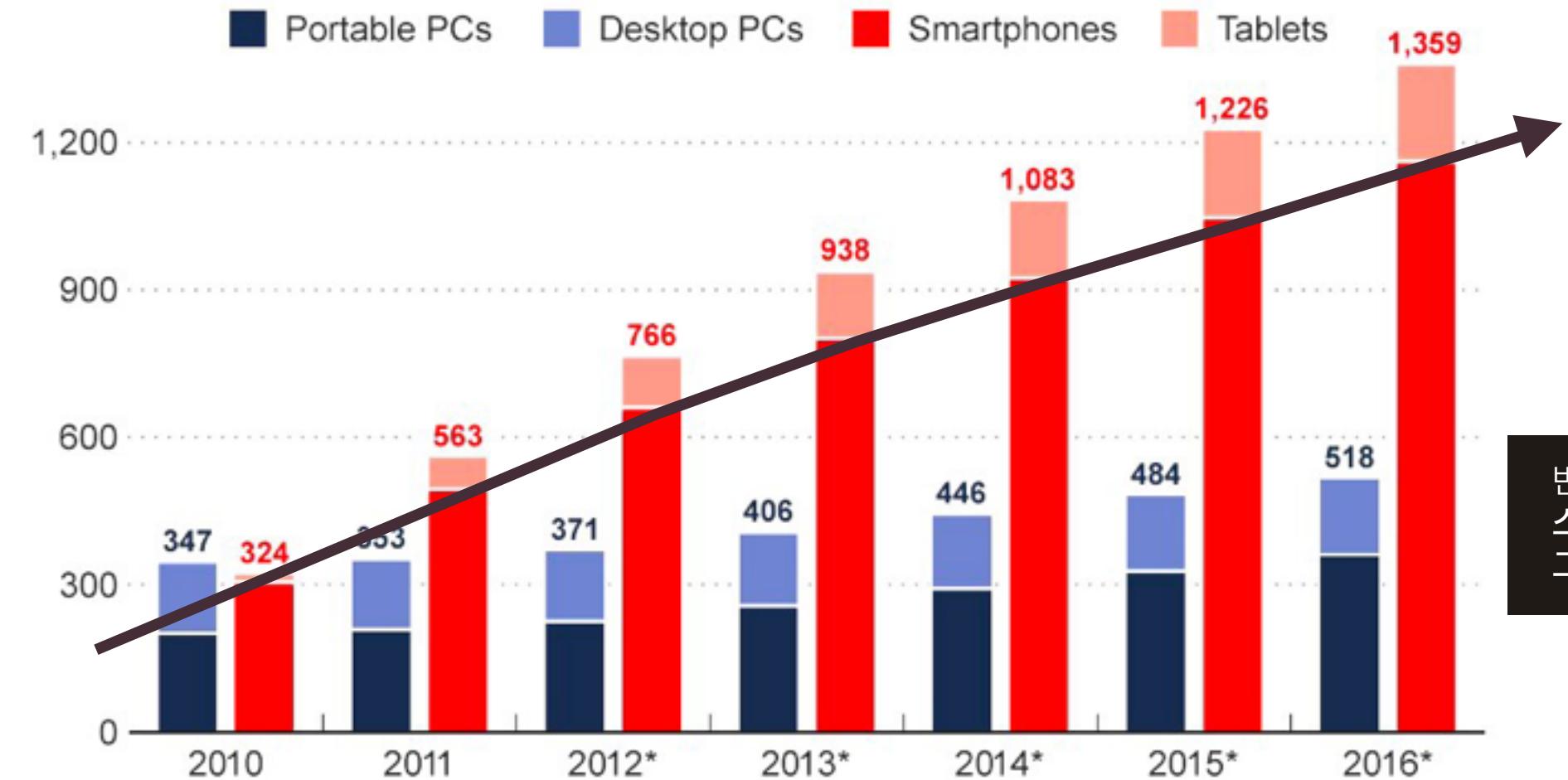
Google  
The New Multi-screen World:  
Understanding Cross-platform Consumer Behavior  
  
August 2012

사람들은 멀티 스크린 기기 중  
스마트폰을 가장 많이 사용하는  
경향을 보인다.

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



반응형 웹 디자인이 등장한 이후,  
스마트폰 환경이 가장 큰 성장 폭을  
그리며 발전한 것을 확인할 수 있다.

# 모바일 사용자 환경을 우선 고려해야

기술을 이용해, 동료를 이용해, 세상을 이용해

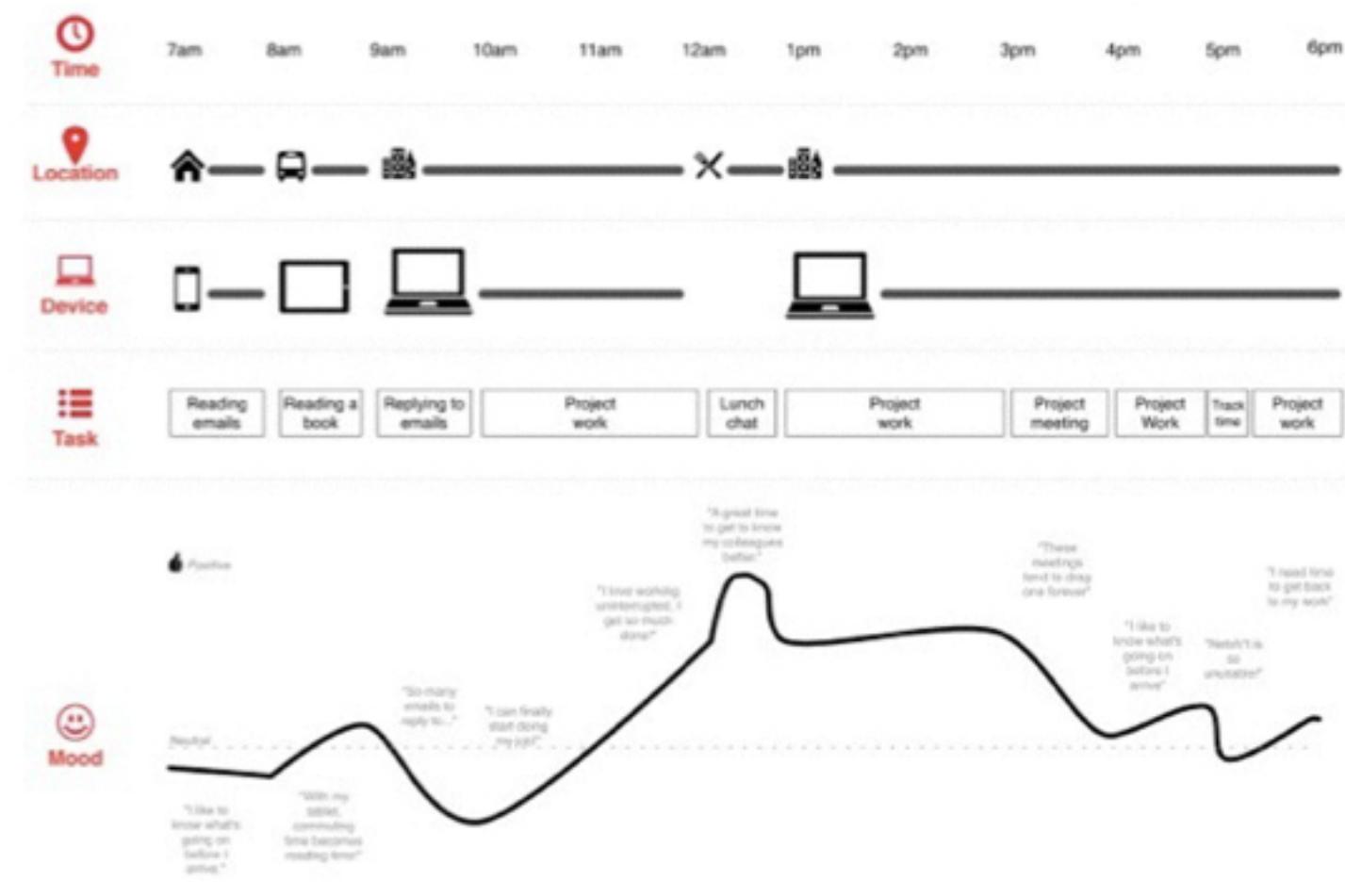
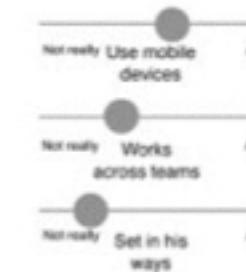


"My days are pretty full, so I try to make the most of out my time."

I am a UI designer and I joined Tigerspike 18 months ago. At work, I need to get "in the zone" to get really productive.

Goals:

- getting as much done as possible and leaving on time.
- sharing my experience with other and learning from them.



사용자 행동 패턴을 분석한 결과  
모바일 환경을 우선하여 서비스를  
제작해야 함을 알 수 있다.

# 반응형 웹 프로젝트를 시작하기 위해 우리가 고려해야 할 것은?

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS



콘텐츠 중심



가급적 네이티브 기능 사용



Form 2.0 도입



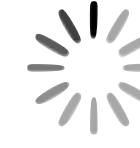
NUI 를 고려한 설계



데이터 커넥션 불안정 고려



콘텐츠를 모호하게 하지 않는 내비게이션



화면에 바로 보이지 않는 콘텐츠는 Lazy Loading



읽기 쉬운 콘텐츠를 구성하는 타이포그래피



SVG 그래픽 및 아이콘 폰트 적극 활용

모바일 퍼스트 전략



성능 고려 (HTTP 요청 회수 줄임)

# 반응형 웹 프로젝트를 시작하기 위해 우리가 고려해야 할 것은?

기술을 이용해, 동료를 이용해, 세상을 이용해

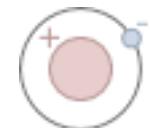
IROPKE Front-End Design CAMPUS



모바일 퍼스트 디자인



디자인 패턴 라이브러리(UI Kit) 제작



웹 리소스 압축 배포 (gZip)



고해상도 디스플레이 대응



프레임워크 도입 검토



마우스 이벤트 핸들링에서 벗어나야 함



구형 IE는 우아한 기능 저하로 대처



애자일(민첩한) 팀웍이 요구됨  
프로젝트 초기부터 협업을 고려하여야 함

모바일 퍼스트 전략

기술을 이용해, 동료를 이용해, 세상을 이용해

Responsive Web Design

RWD

WEB 캠퍼스 디자인

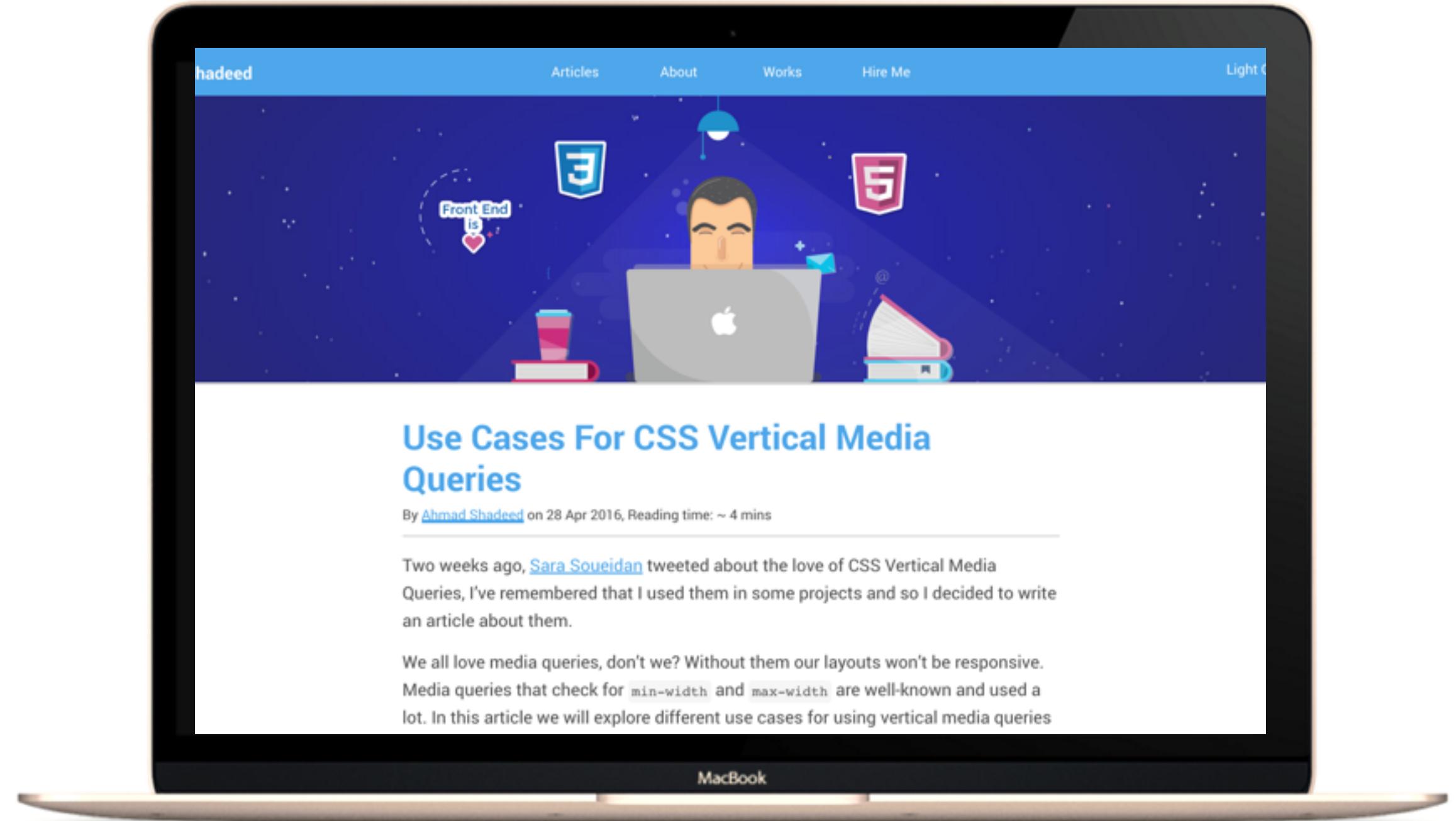


## 「반응형 웹 디자인」 레이아웃

미디어쿼리 수평 방향(min-width, max-width)에 대해서는 많은 사례가 있지만, 수직 방향(min-height, max-height)에 대한 사례를 찾기란 쉽지 않다. 수직 방향에 대해 고민해보자.



기술을 이용해, 동료를 이용해, 세상을 이용해



<https://ishadeed.com/article/vertical-media-queries/>

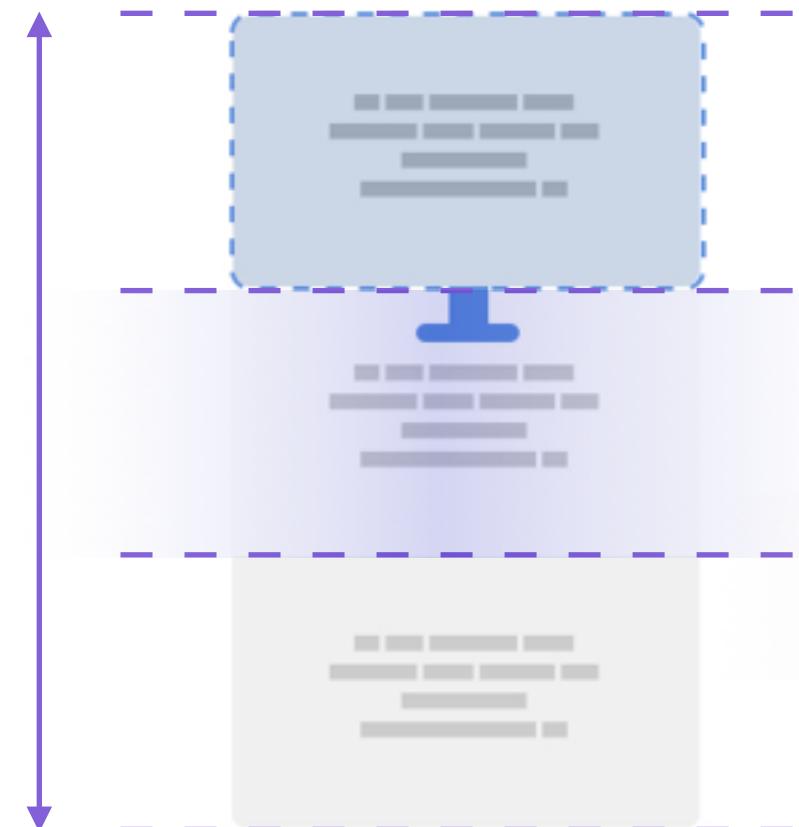
Sectioning Content

# 섹션 콘텐츠

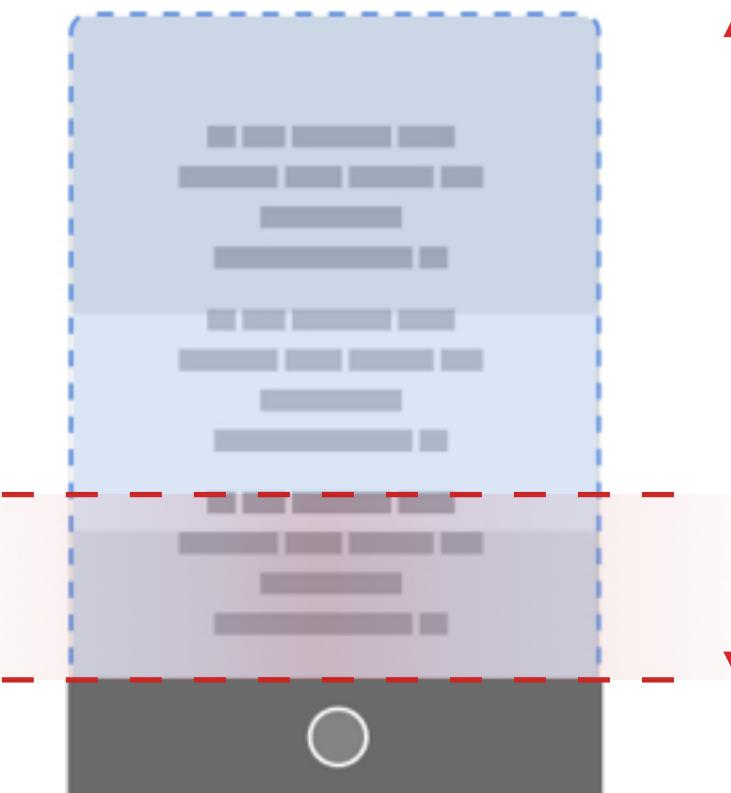
기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

뷰포트 높이가 클 경우,  
섹션 콘텐츠 높이를 뷰포트  
높이만큼 설정



Good



Bad

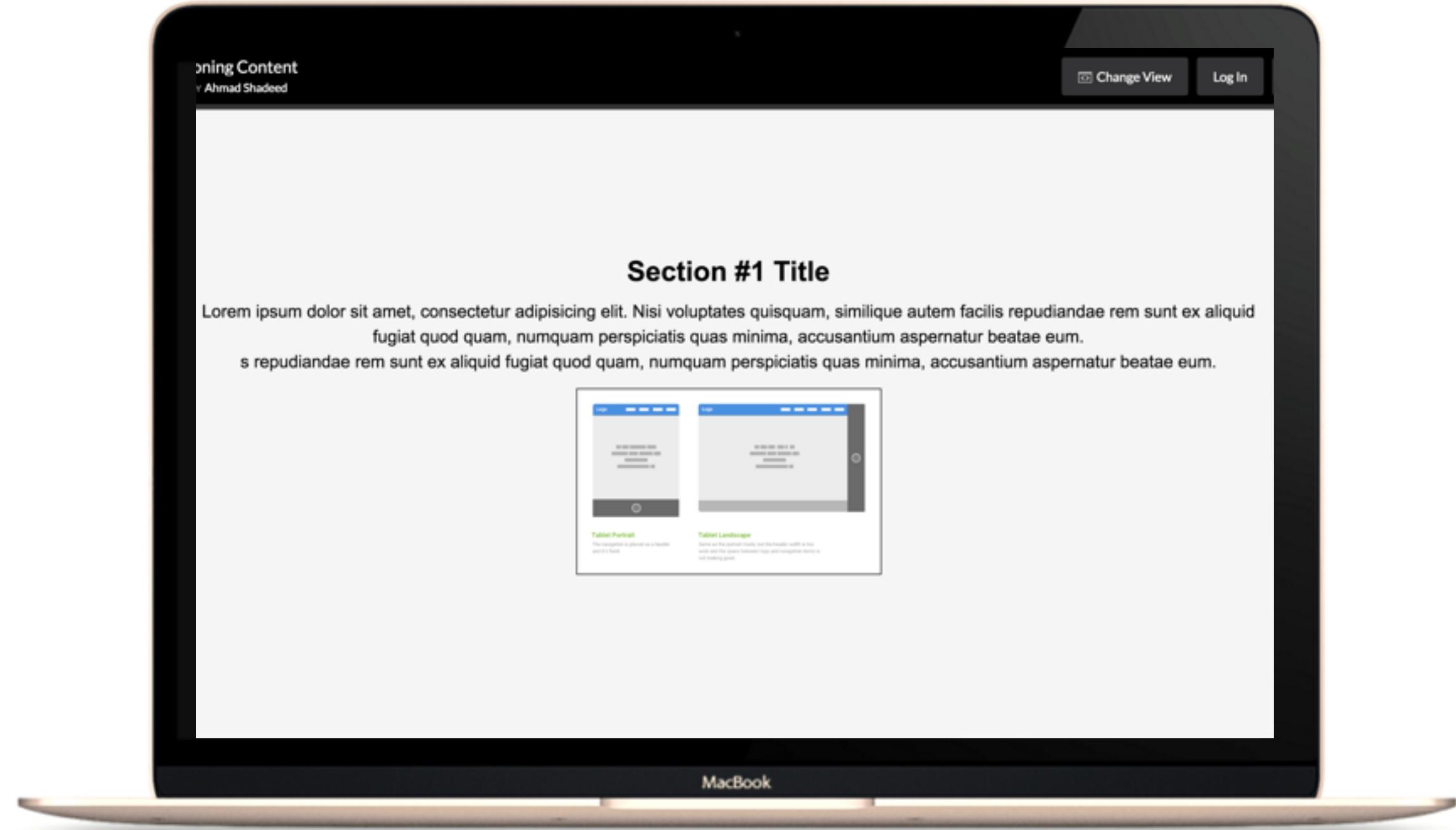
Sectioning Content

## 섹션ning 콘텐츠

# 뷰포트 높이 설정

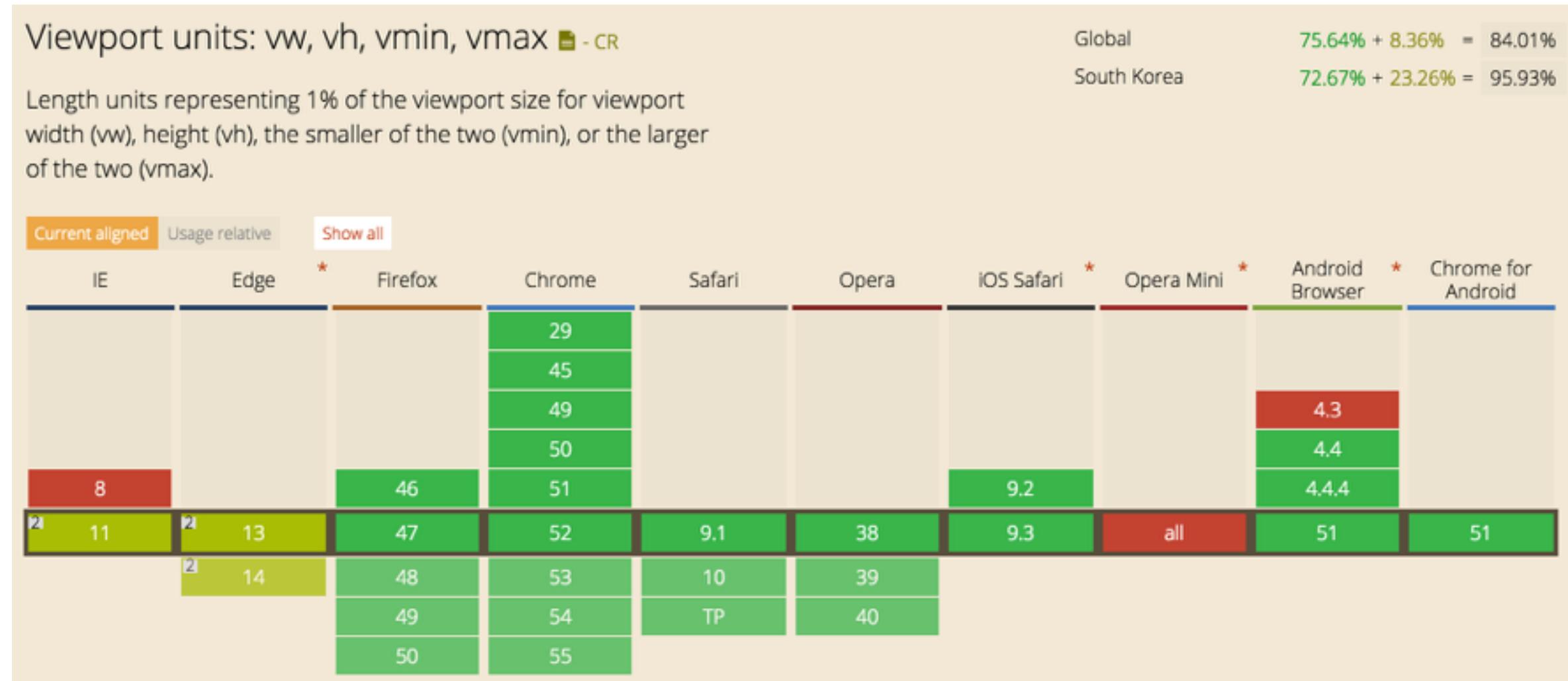
뷰포트 높이가 큰 조건이 충족되면

```
@media (min-height: 400px) {  
    .section {  
        height: 100vh; ←  
    }  
}
```



기술을 이용해, 동료를 이용해, 세상을 이용해

## 브라우저 호환성 테이블



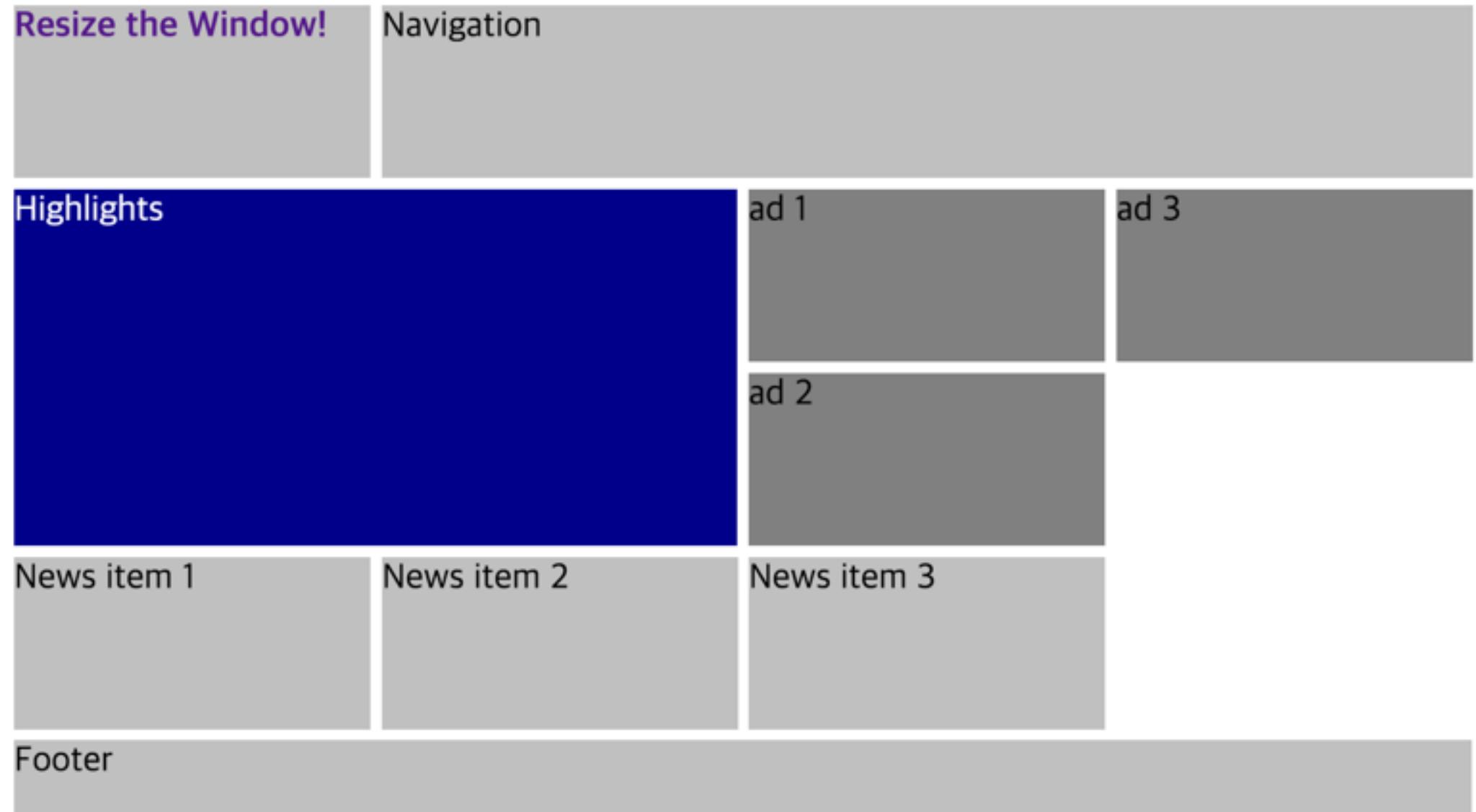
뷰포트 크기에 따라 상대적인 단위(vw)를 글자 크기에 적용한 예

The screenshot shows a code editor interface with three tabs: HTML, CSS, and JS. The HTML tab contains the code <h1>I resize.</h1>. The CSS tab contains the code h1 { font-size: 20vw; }. The JS tab is currently empty.

```
❶ <h1>I resize.</h1>
❶ h1 {
❷   font-size: 20vw;
❸ }
```



<http://fittextjs.com/>



vminpoly

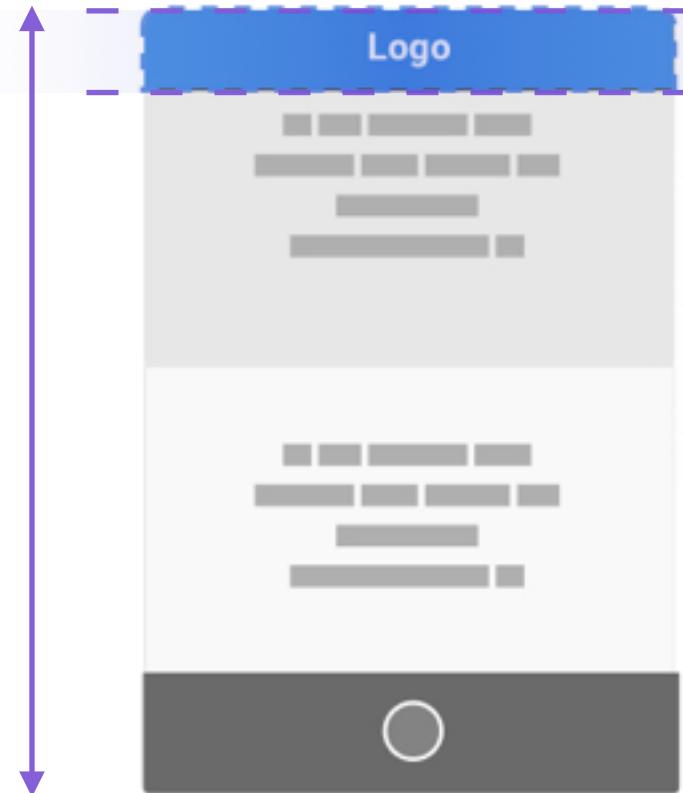
A polyfill for CSS units vw, vh & vmin.



Fixed Header

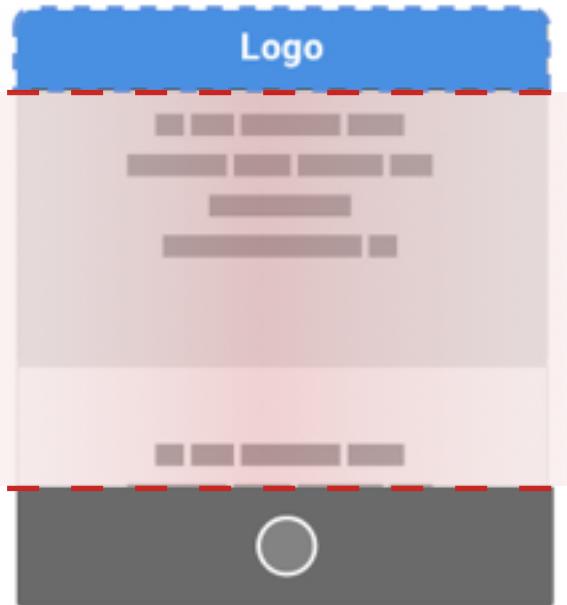
# 헤더 고정

뷰포트 높이가 클 경우  
헤더 고정 필요



Good

뷰포트 높이가 특정 길이보다  
작아진다면 헤더 고정은 불필요



Bad

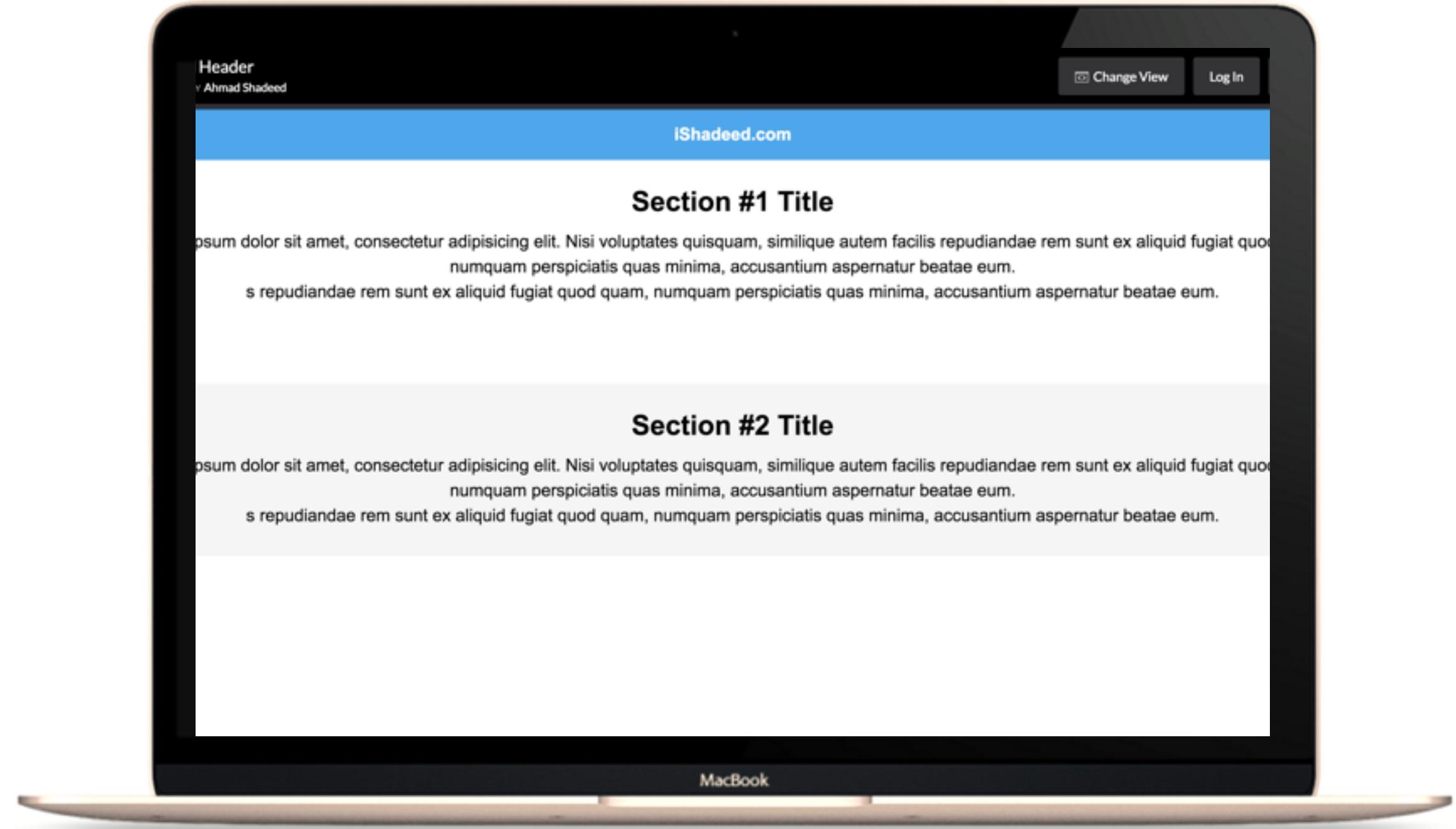
Fixed Header

헤더 고정

# 고정기능추가

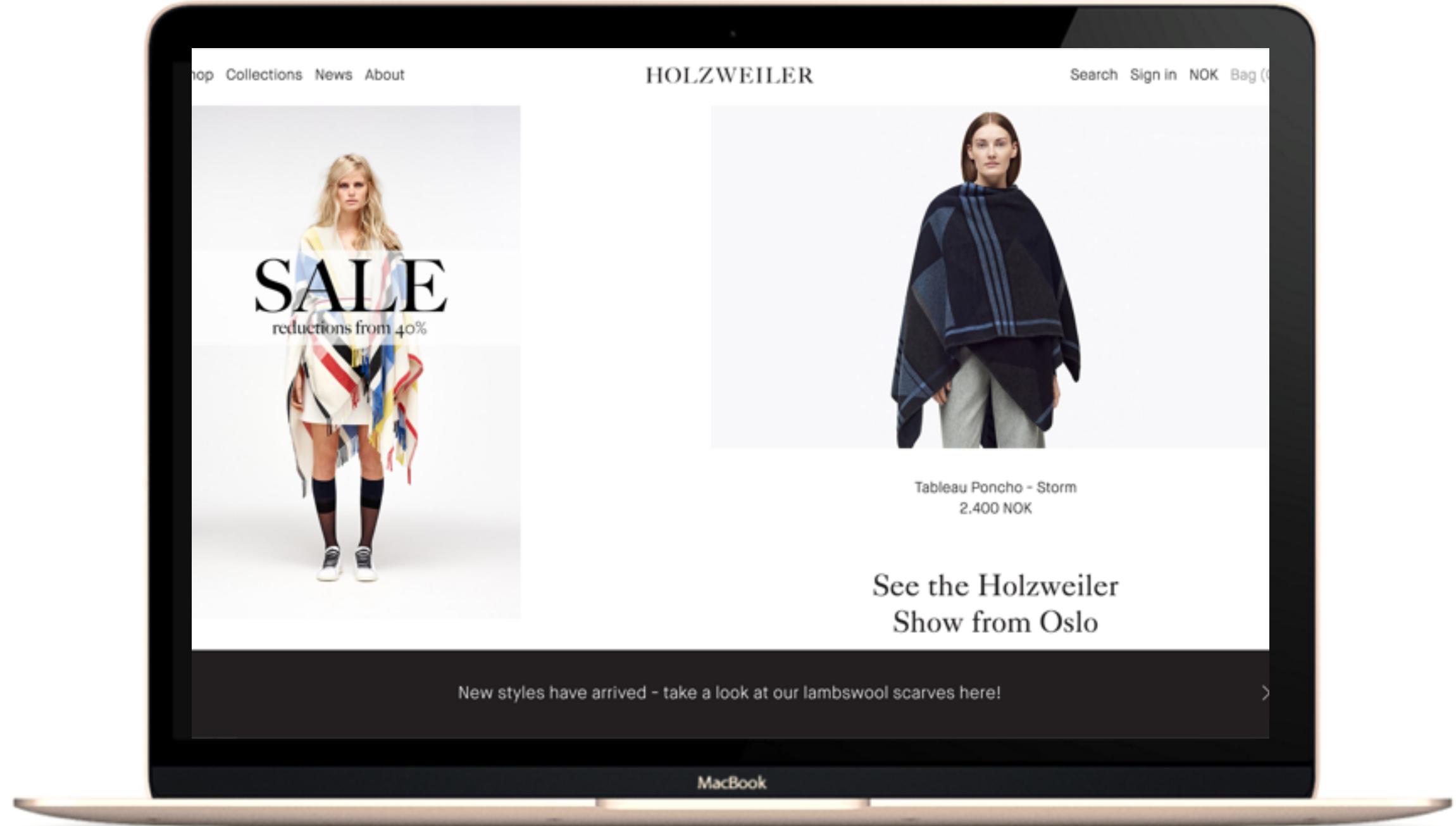
뷰포트 높이가 큰 조건이 충족되면

```
@media (min-height: 600px) {  
    .site-header {  
        position: fixed; ←  
        /* ... */  
    }  
}
```



기술을 이용해, 동료를 이용해, 세상을 이용해

기술을 이용해, 동료를 이용해, 세상을 이용해



<https://www.holzweiler.no/>

Fixed Footer

## 푸터 고정

기술을 이용해, 동료를 이용해, 세상을 이용해



헤더와 마찬가지로 뷰포트 높이가 특정 길이보다  
작아진다면 푸터 고정은 불필요



Good



Bad

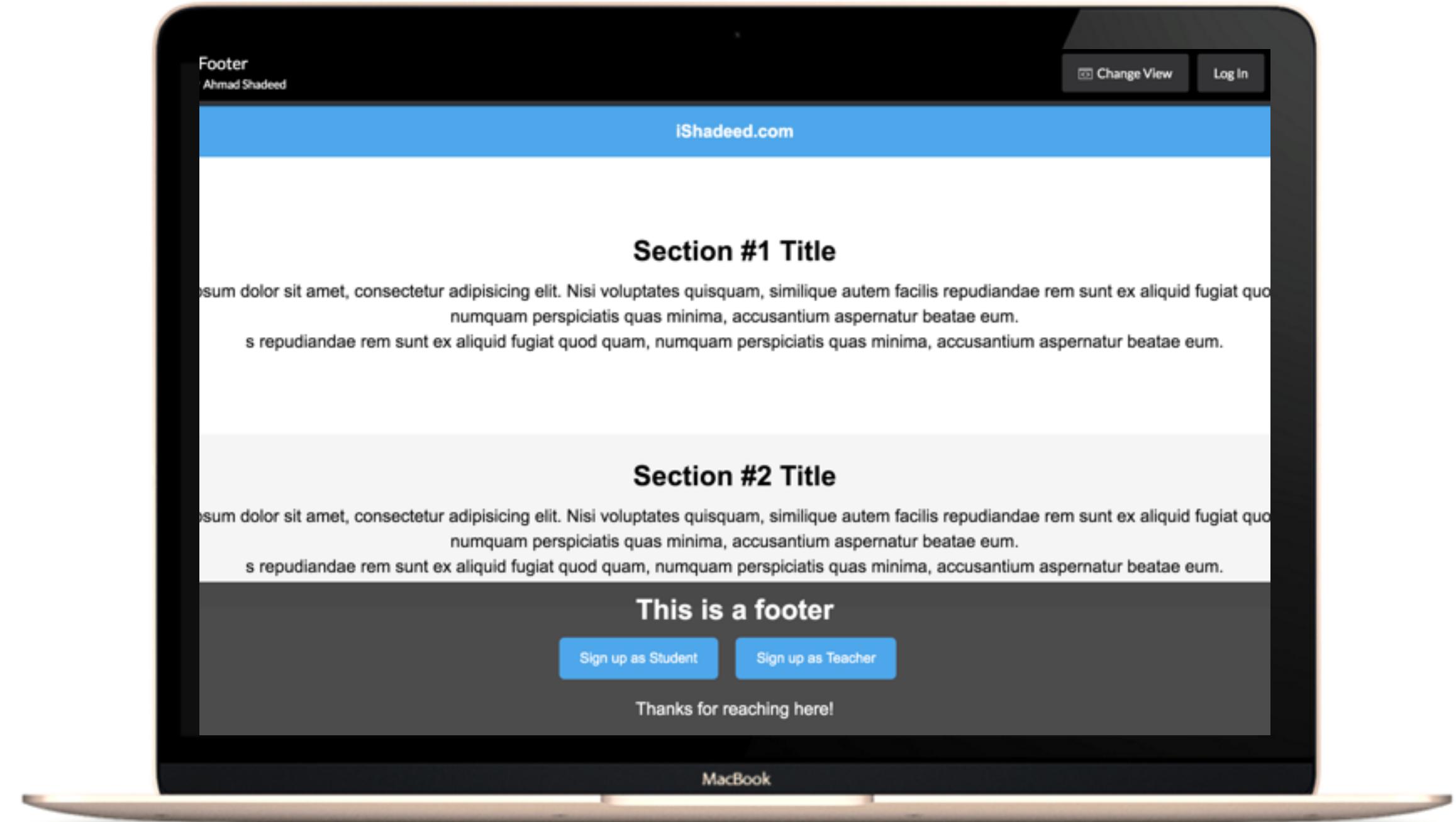
Fixed Footer

푸터 고정

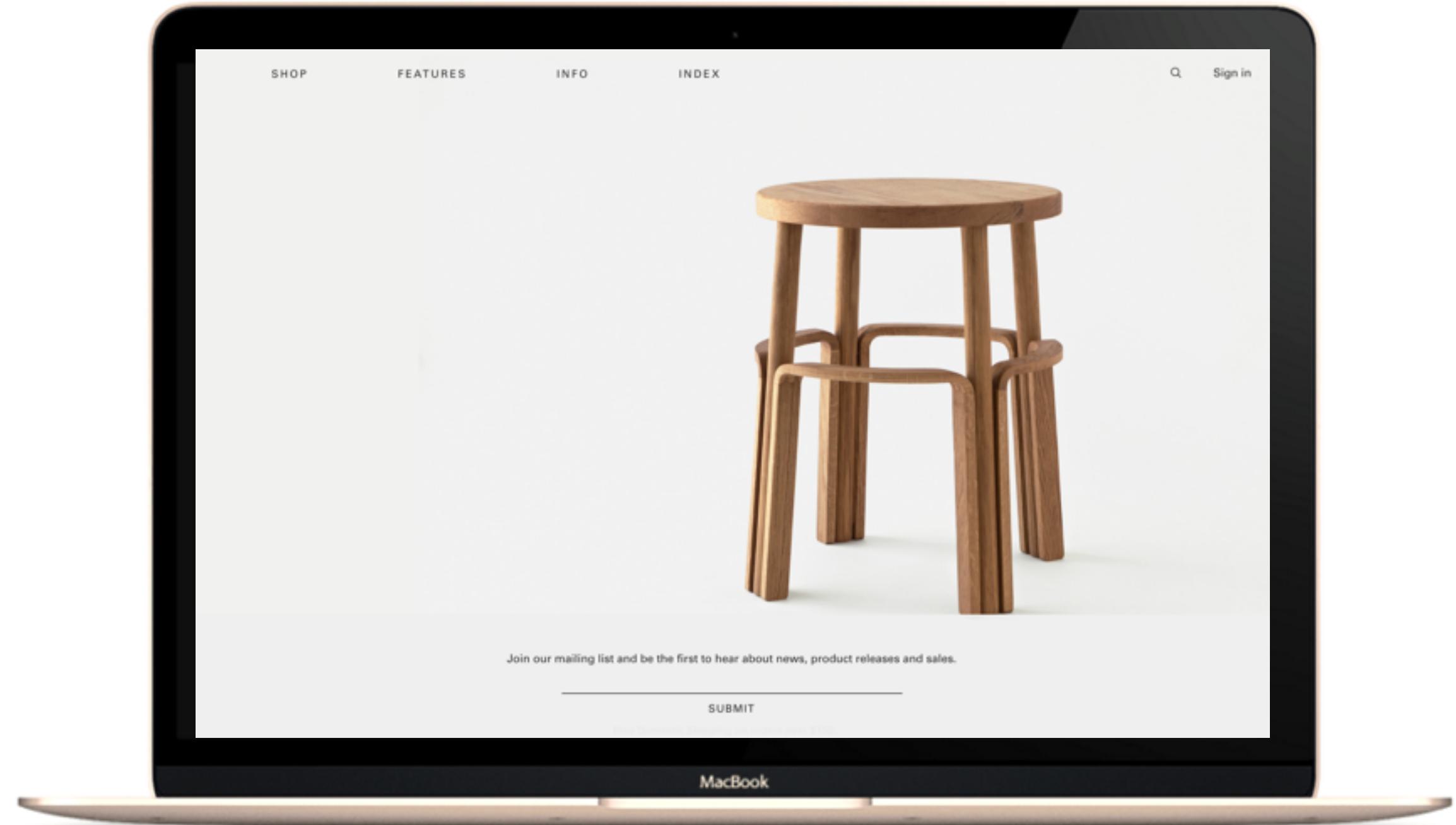
# 고정기능추가

뷰포트 높이가 큰 조건이 충족되면

```
@media (min-height: 800px) {  
    .site-footer {  
        position: fixed; ←  
        /* ... */  
    }  
}
```



기술을 이용해, 동료를 이용해, 세상을 이용해



<https://makr.com/>

Modals

## 모달 윈도우(레이어 팝업)

기술을 이용해, 동료를 이용해, 세상을 이용해



뷰포트 높이가 좁은 화면의 경우  
화면 높이 가운데 배치될 경우  
화면에서 잘릴 위험이 있다



Good



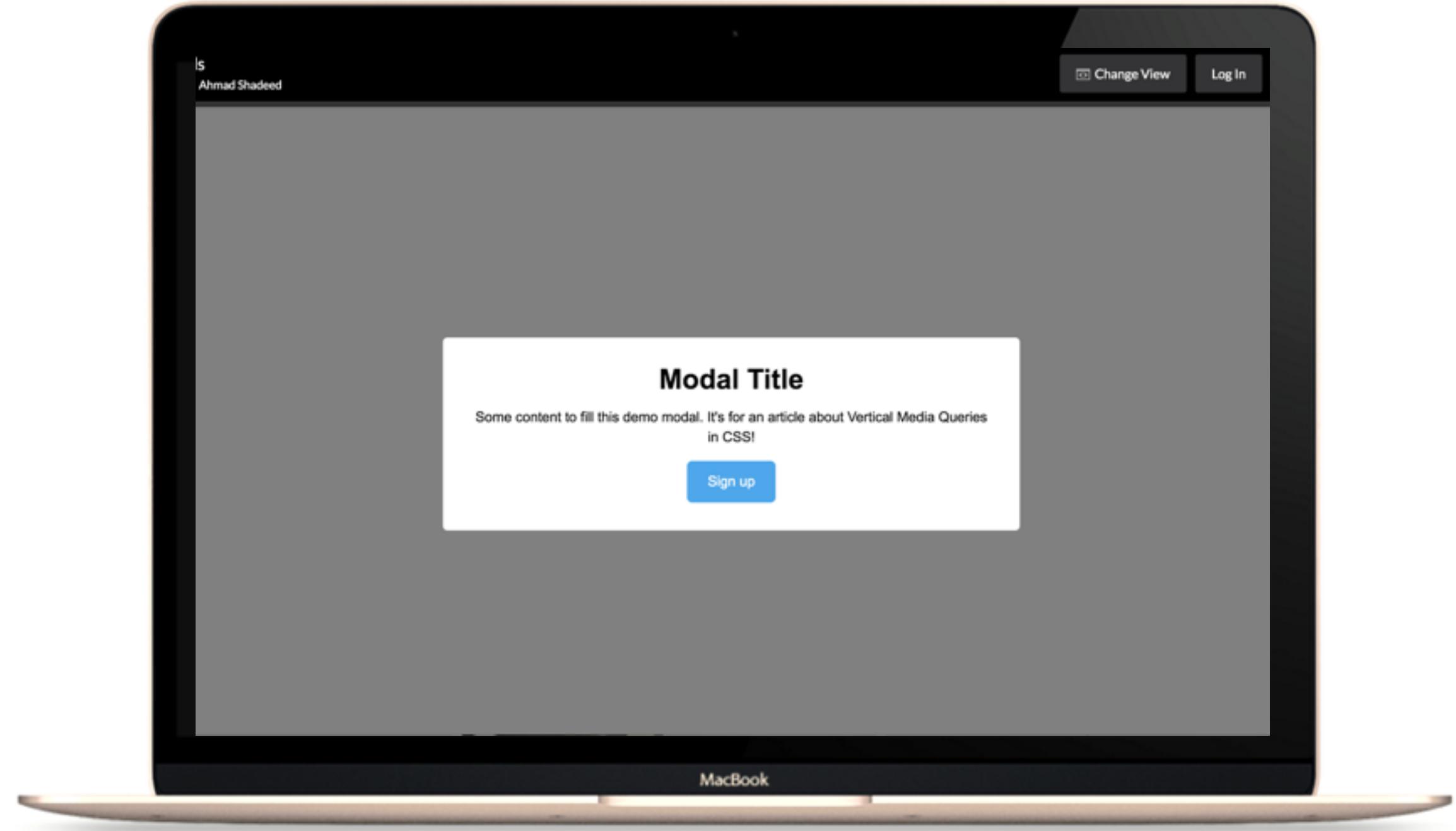
Bad

Modals

# 모달 윈도우(레이어 팝업)

기술을 이용해, 동료를 이용해, 세상을 이용해

```
.modal {  
  position: absolute;  
  left: 50%;  
  top: 30px; /* 작은 모바일 화면 위에서부터 위치 시킴 */ ←  
  transform: translate(-50%, 0);  
}  
  
@media (min-height: 800px) {  
  .modal {  
    top: 50%; /* 큰 화면에서는 화면 높이 가운데 위치 시킴 */  
    transform: translate(-50%, -50%);  
  }  
}
```

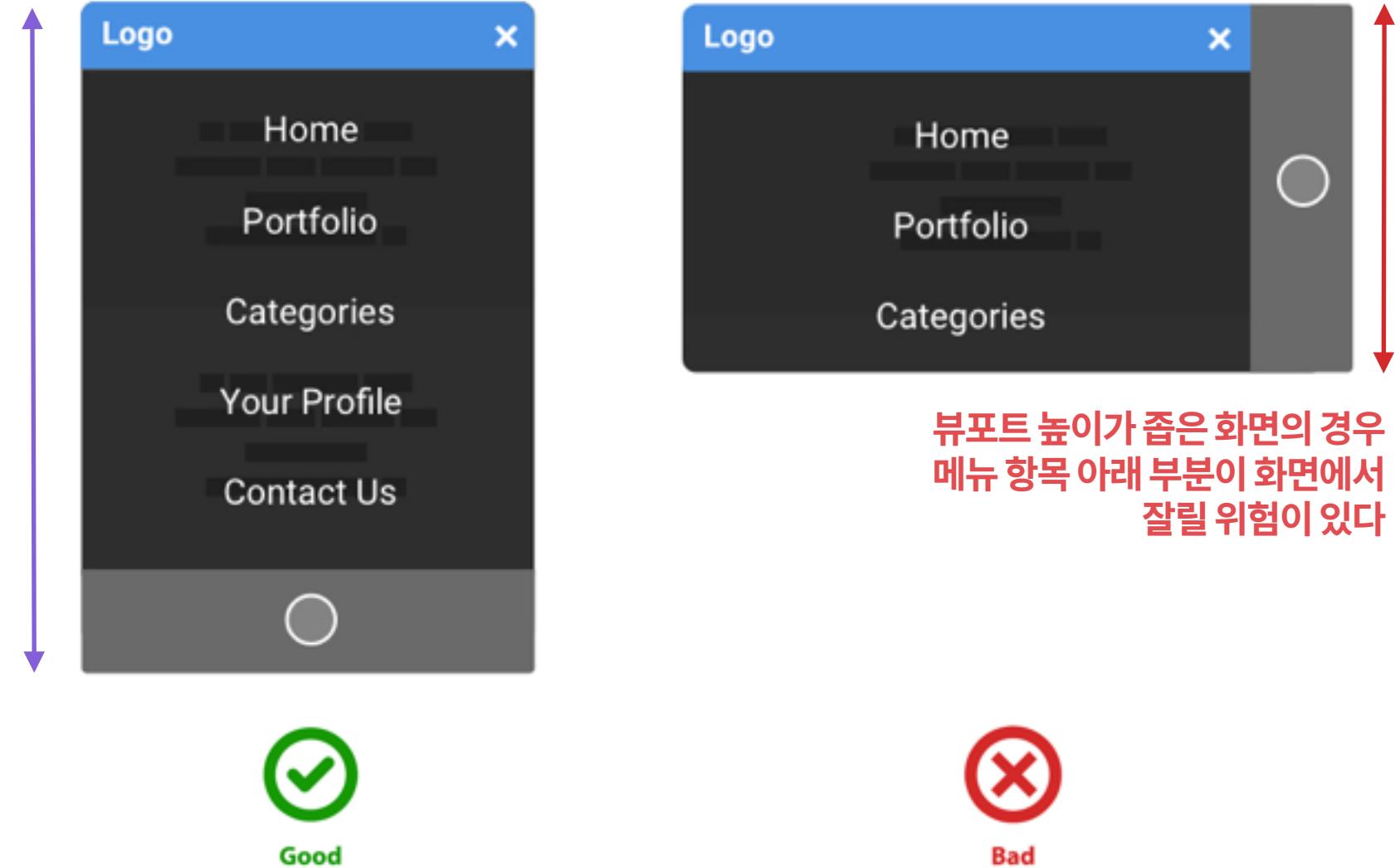


기술을 이용해, 동료를 이용해, 세상을 이용해

<http://codepen.io/shadeed/full/c3d9d8b94567624bec235f3512304f98/>

Mobile Menu

모바일 메뉴



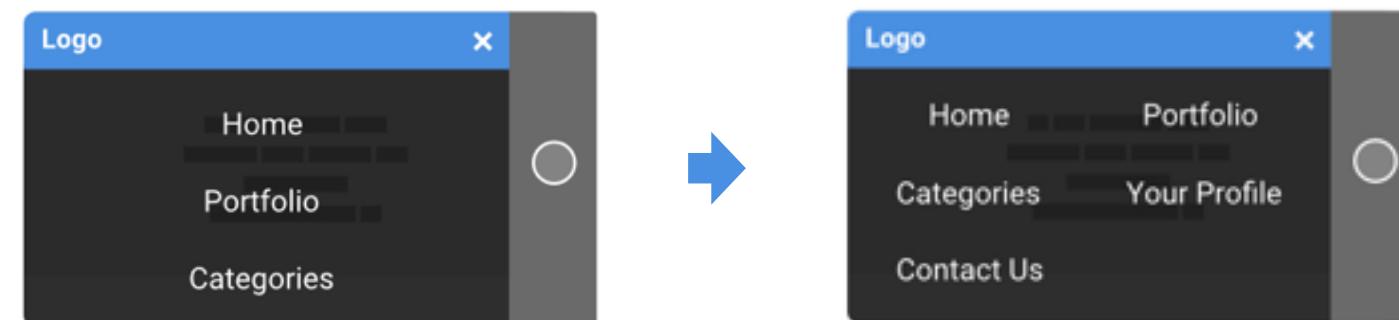
Mobile Menu

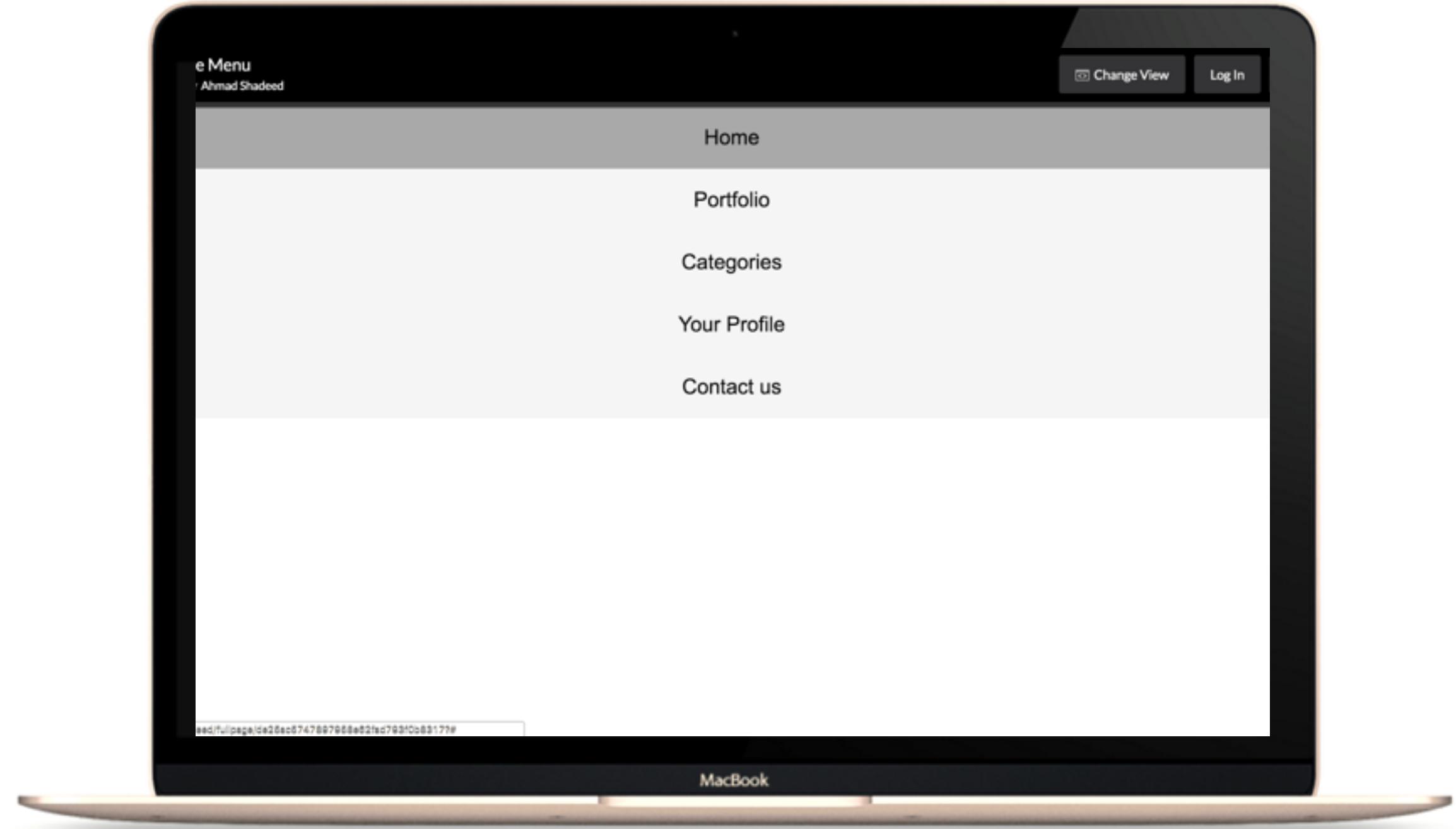
모바일 메뉴

# 플로팅초기화

뷰포트 높이가 커지면, 모바일 메뉴 설정

```
@media (min-height: 400px) {  
    .nav-item {  
        float: none; ←  
        width: 100%;  
    }  
}
```





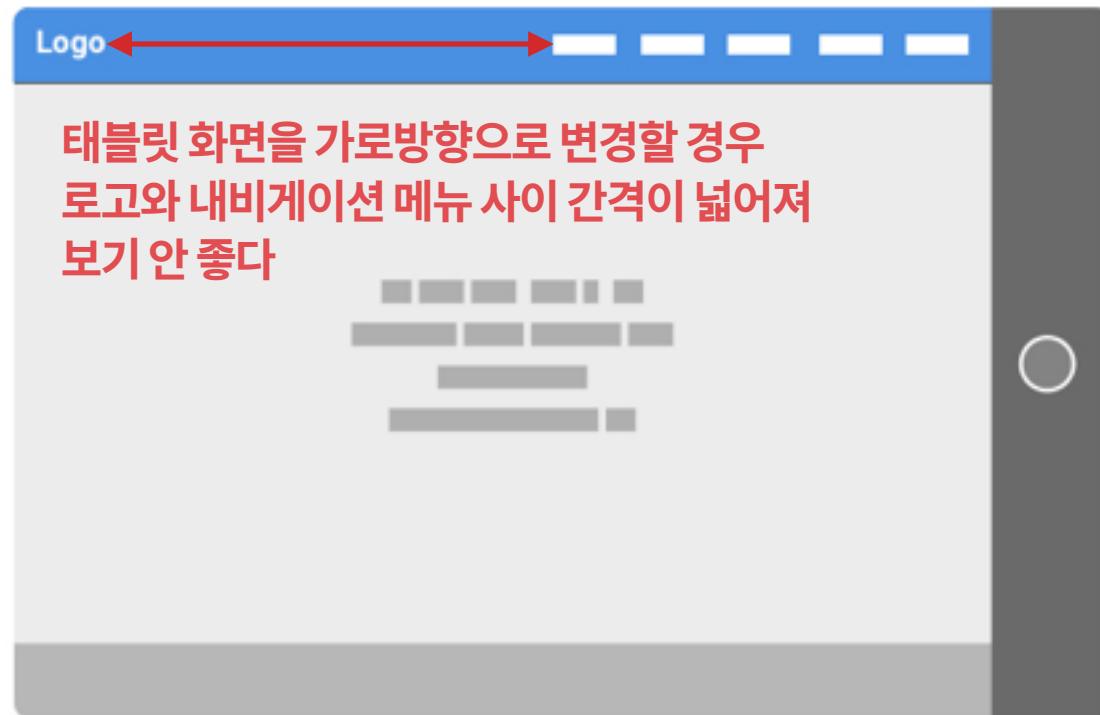
기술을 이용해, 동료를 이용해, 세상을 이용해

<http://codepen.io/shadeed/full/de25ac5747897958e62fad793f0b8317/>

Sidebar VS Header

# 사이드바 vs 헤더

기술을 이용해, 동료를 이용해, 세상을 이용해



Good

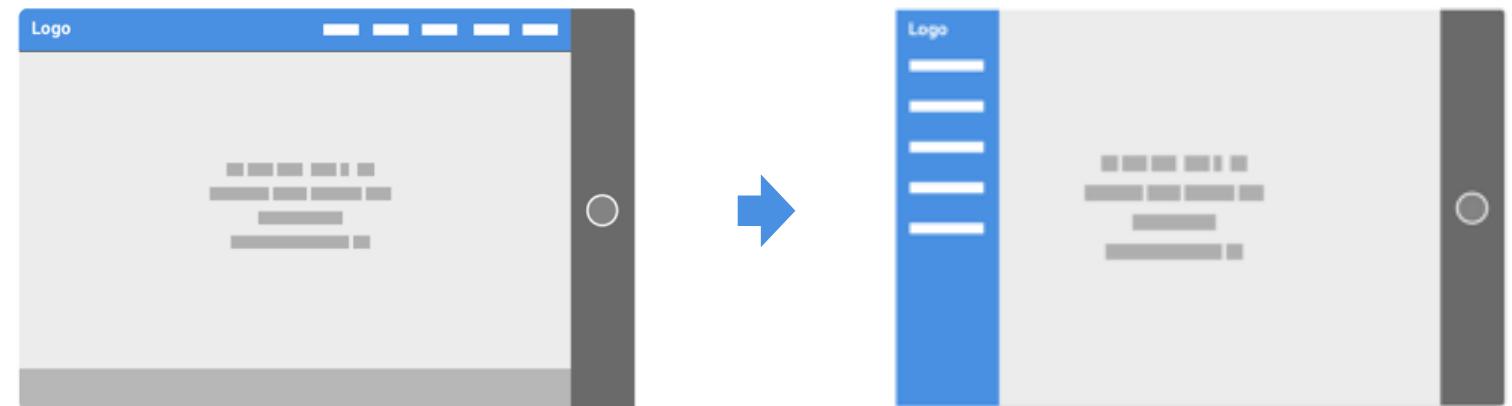


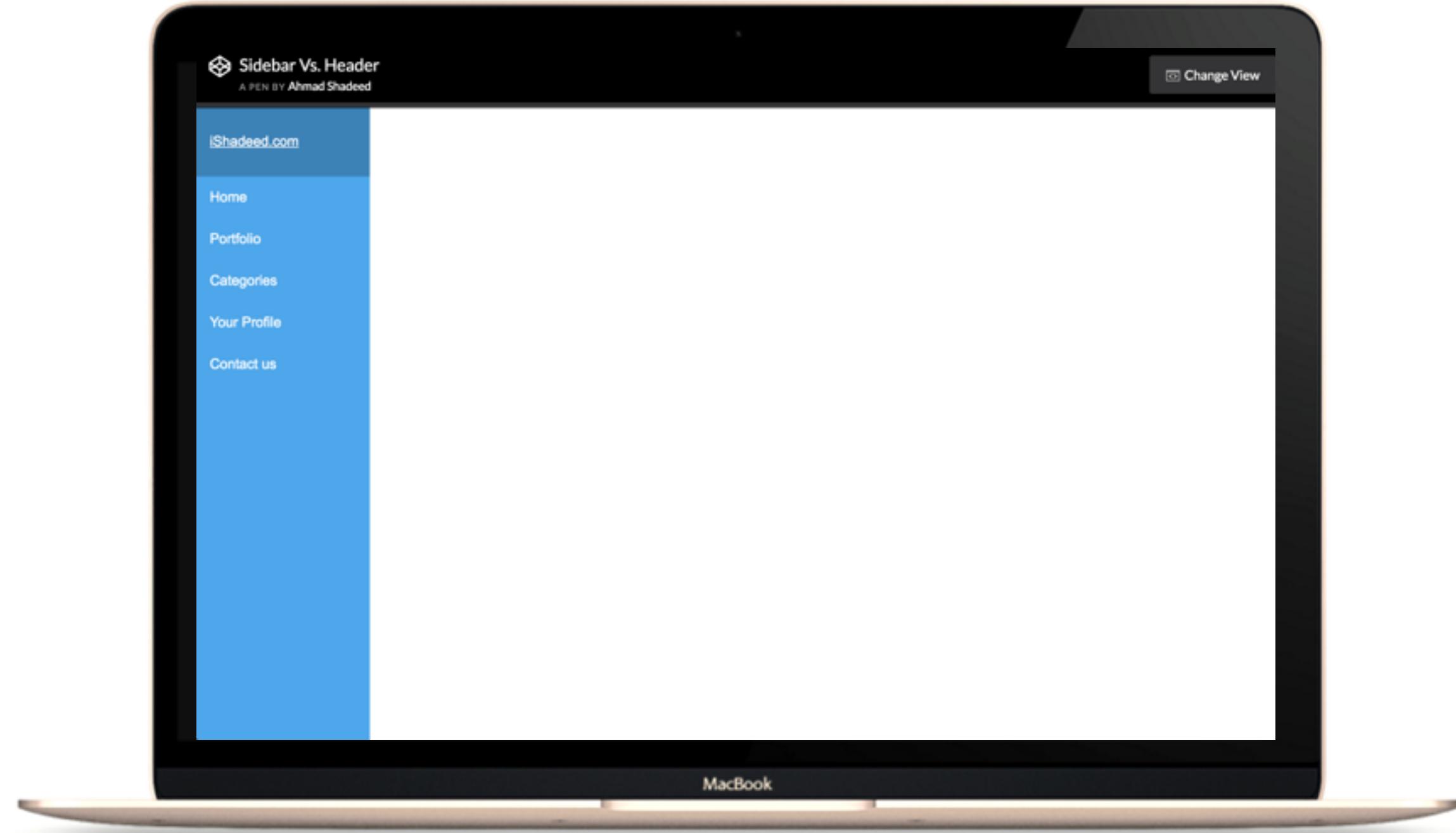
Bad

Sidebar VS Header

# 사이드바 vs 헤더

```
@media (min-width: 1000px) and (min-height: 700px) {  
    .header {  
        // 헤더를 사이드 메뉴로 변경  
        position: fixed;  
        left: 0;  
        top: 0;  
        bottom: 0;  
        /* ... */  
    }  
}
```



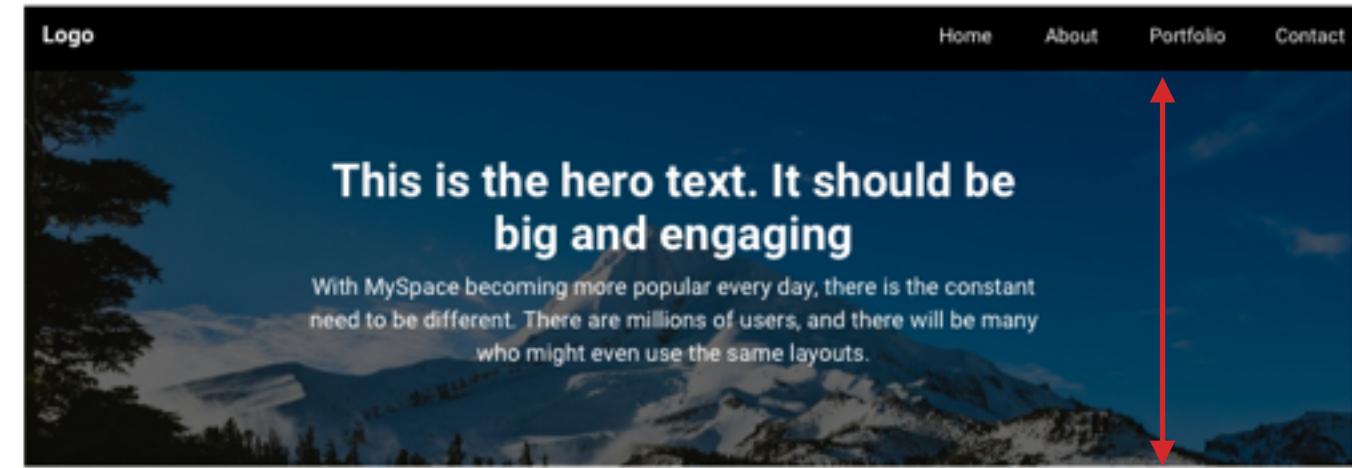
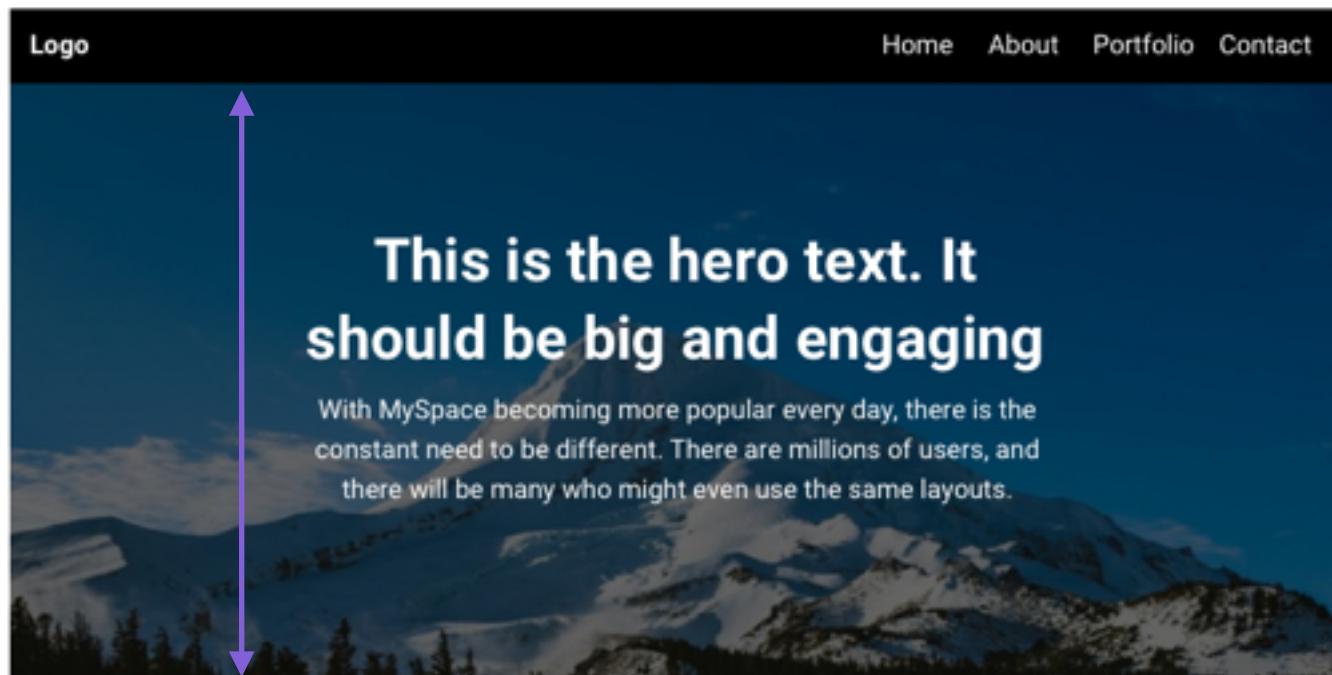


<http://codepen.io/shadeed/full/d5af72aa8816318dbc8977dfe54c31f9/>

Shrinking Font Sizes and Spacing

# 글자 크기 및 간격 축소

기술을 이용해, 동료를 이용해, 세상을 이용해



뷰포트 높이가 좁은 화면의 경우  
글자 크기 및 간격의 변경이 필요하다



Good



Bad

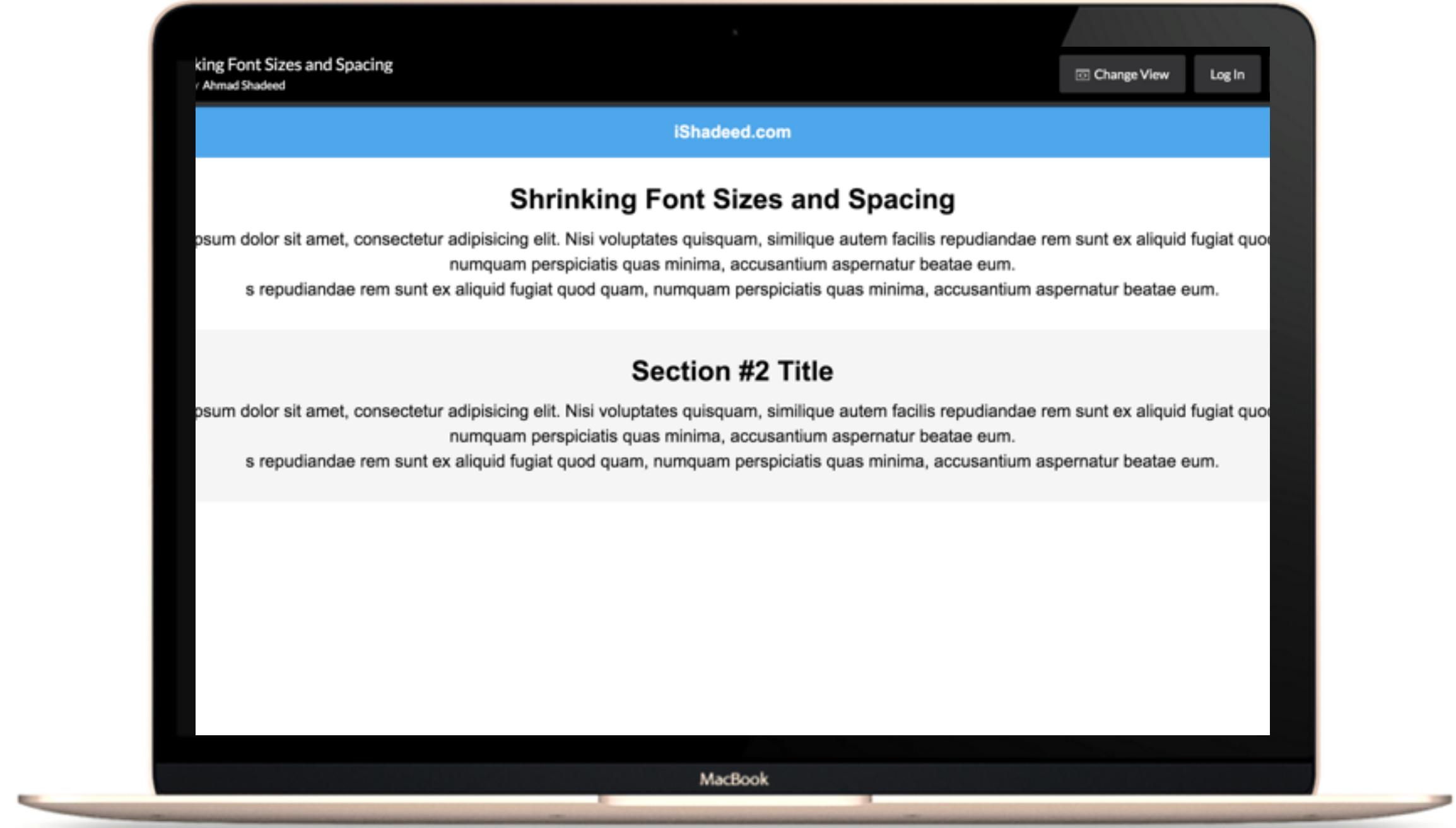
Shrinking Font Sizes and Spacing

글자 크기 및 간격 축소

# 글/자/크/기/변/경

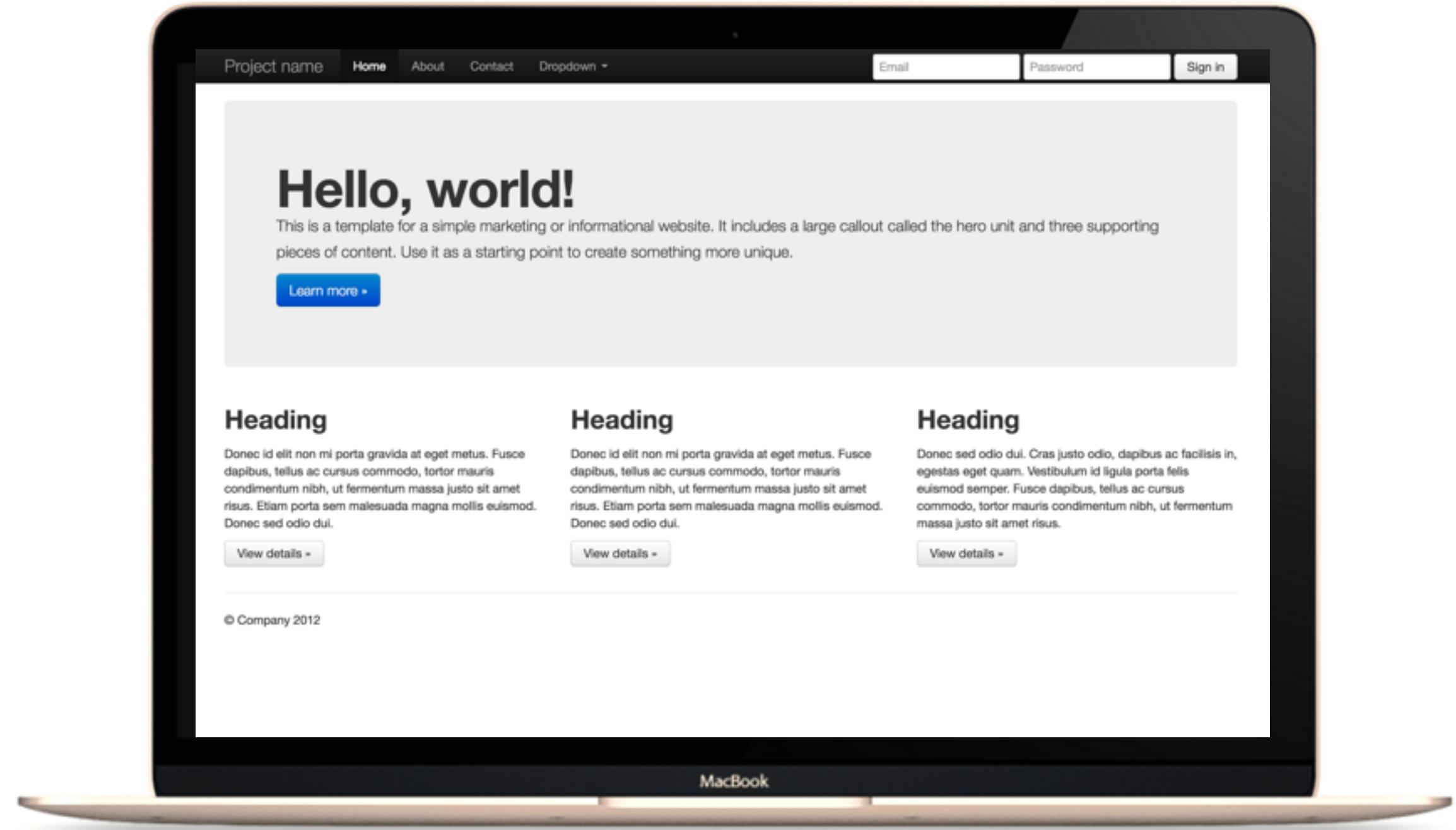
뷰포트 높이가 큰 조건이 충족되면

```
section {  
    font-size: 0.8125rem;  
}  
  
@media (min-height: 400px) {  
    section {  
        font-size: 1rem;  
    }  
}
```

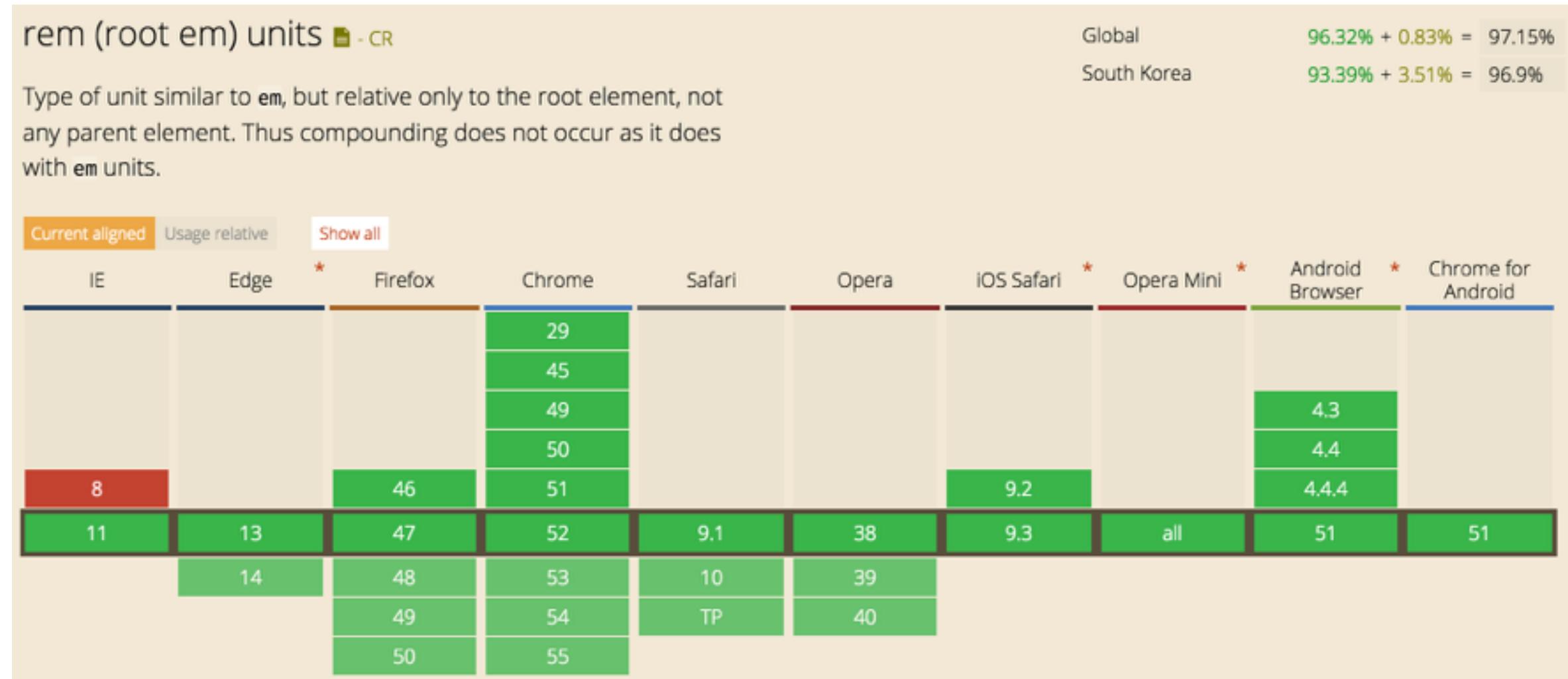


기술을 이용해, 동료를 이용해, 세상을 이용해

<http://codepen.io/shadeed/full/4548a90488caef737f055b5cf28ad44c/>



## 브라우저 호환성 테이블



## em, rem 단위 사용 방법

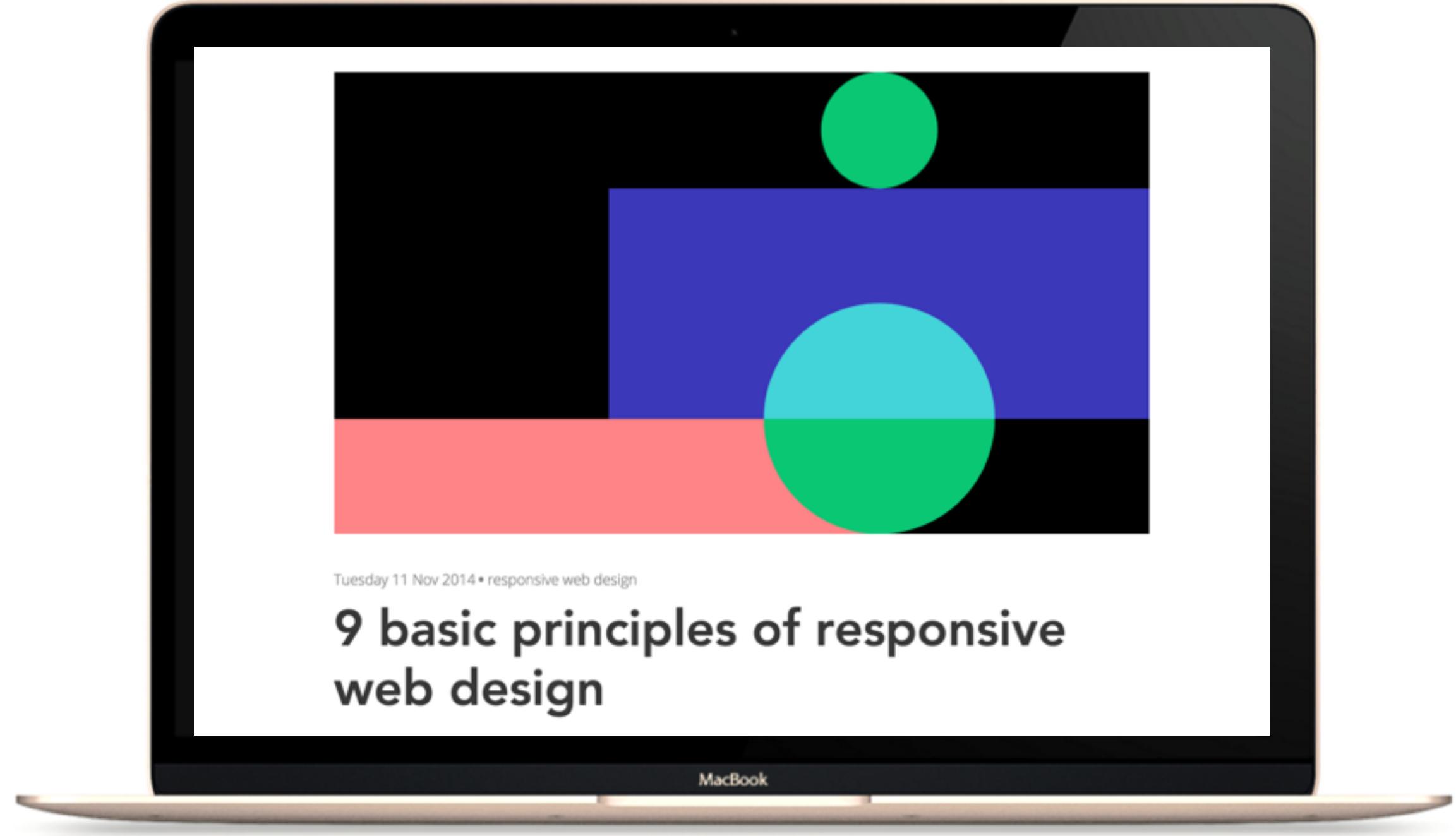


기술을 이용해, 동료를 이용해, 세상을 이용해

Responsive Web Design  
**RWD** 웹사이트 관리



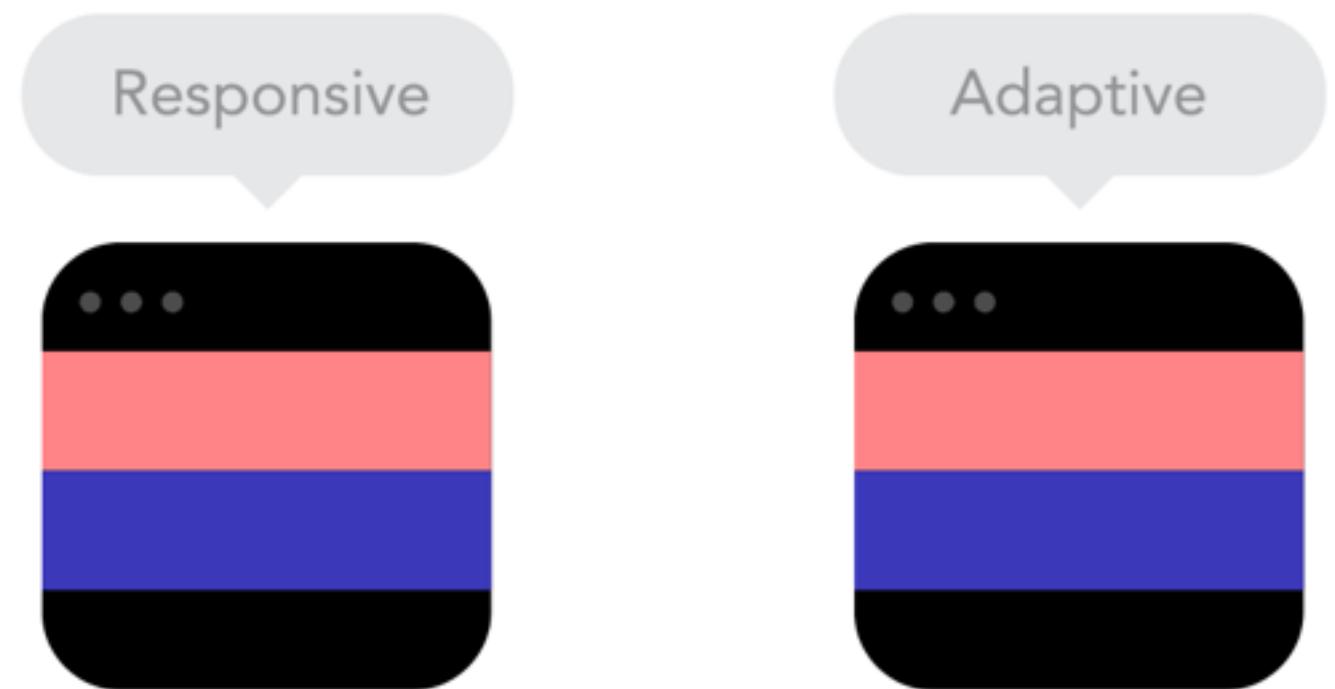
기술을 이용해, 동료를 이용해, 세상을 이용해



<http://blog.froont.com/9-basic-principles-of-responsive-web-design/>

# 반응 VS 적응 / 레이아웃 /

기술을 이용해, 동료를 이용해, 세상을 이용해



# /일/반/하/름/



기술을 이용해, 동료를 이용해, 세상을 이용해

# 유동 VS 고정 /단위/

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

Relative Units



Static Units



# /중/단/점/

기술을 이용해, 동료를 이용해, 세상을 이용해

With Breakpoints

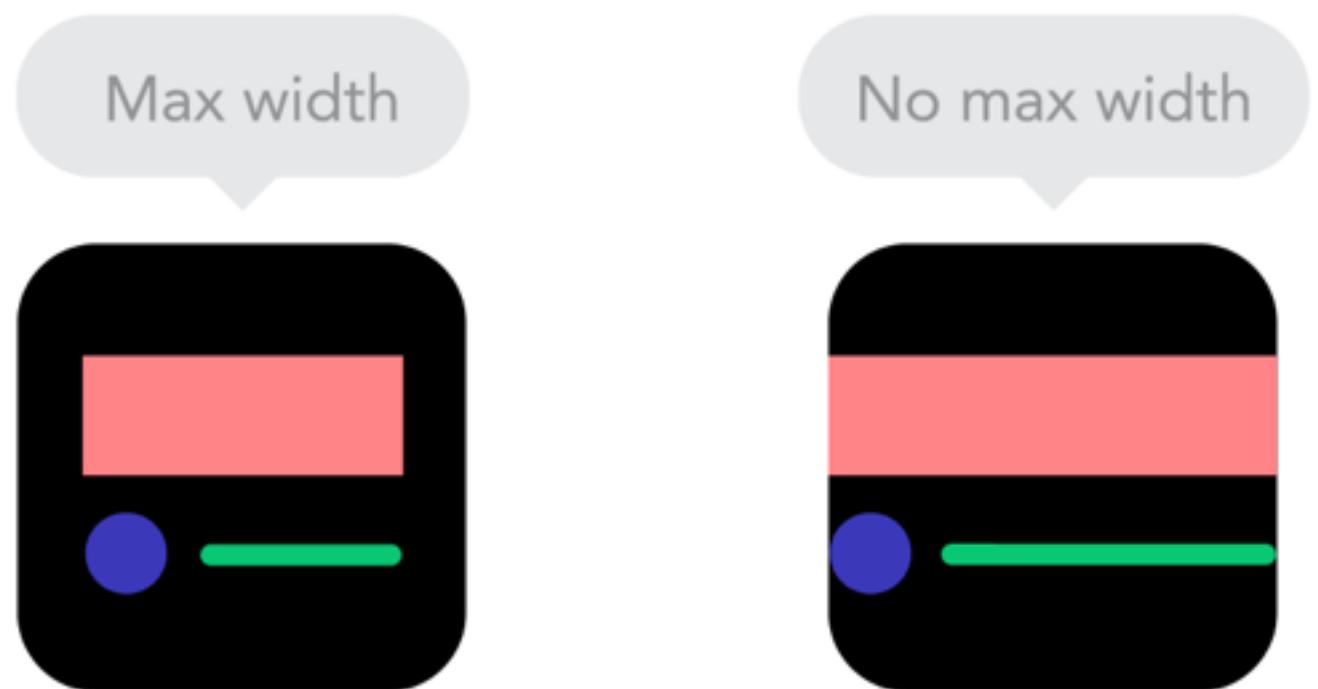


Without Breakpoints



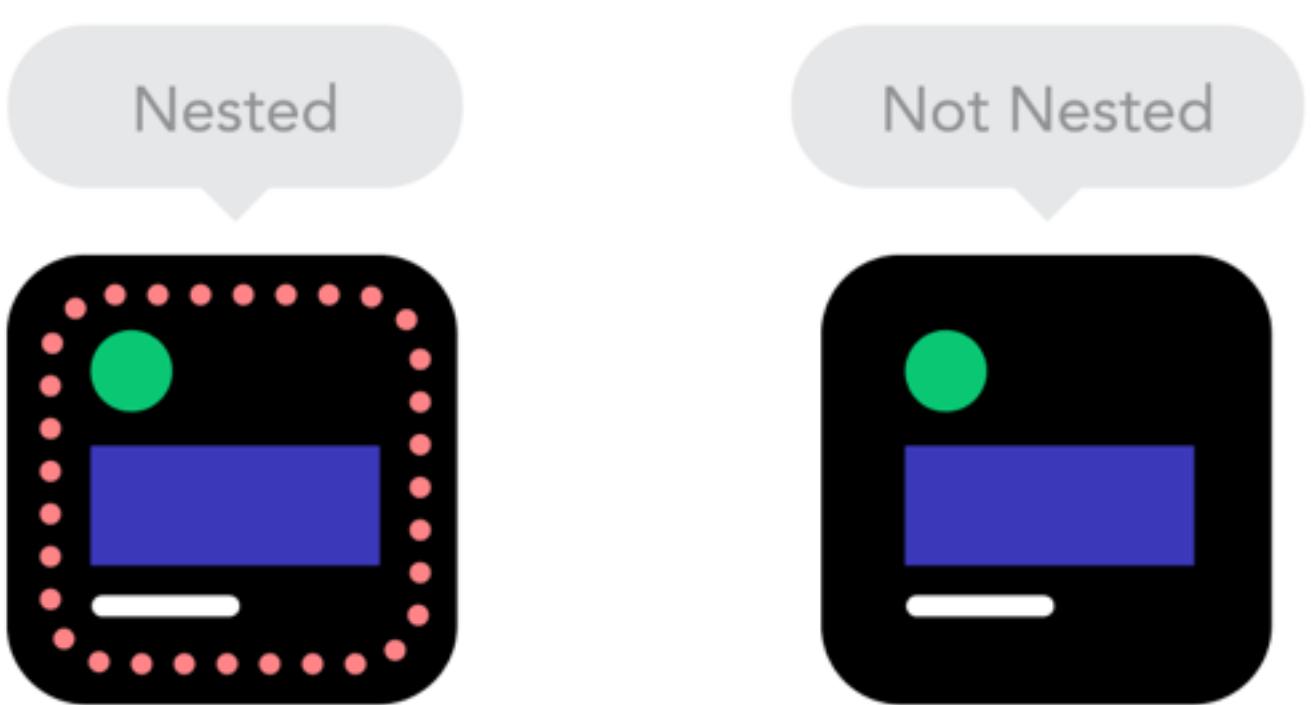
# /최/대/가/로/폭/

기술을 이용해, 동료를 이용해, 세상을 이용해



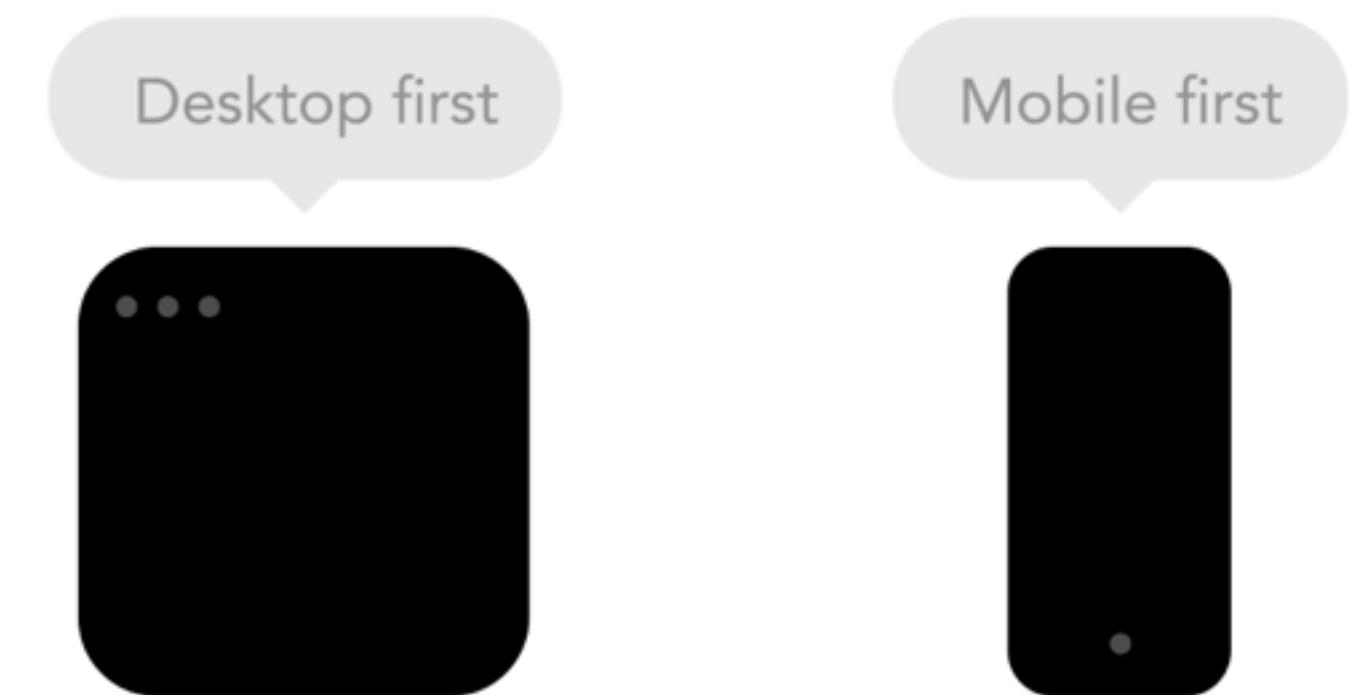
# /중/첩/된/

기술을 이용해, 동료를 이용해, 세상을 이용해



# 데스크탑 VS 모바일 퍼스트

기술을 이용해, 동료를 이용해, 세상을 이용해



# 시스템 VS 웹 폰트

기술을 이용해, 동료를 이용해, 세상을 이용해

System fonts

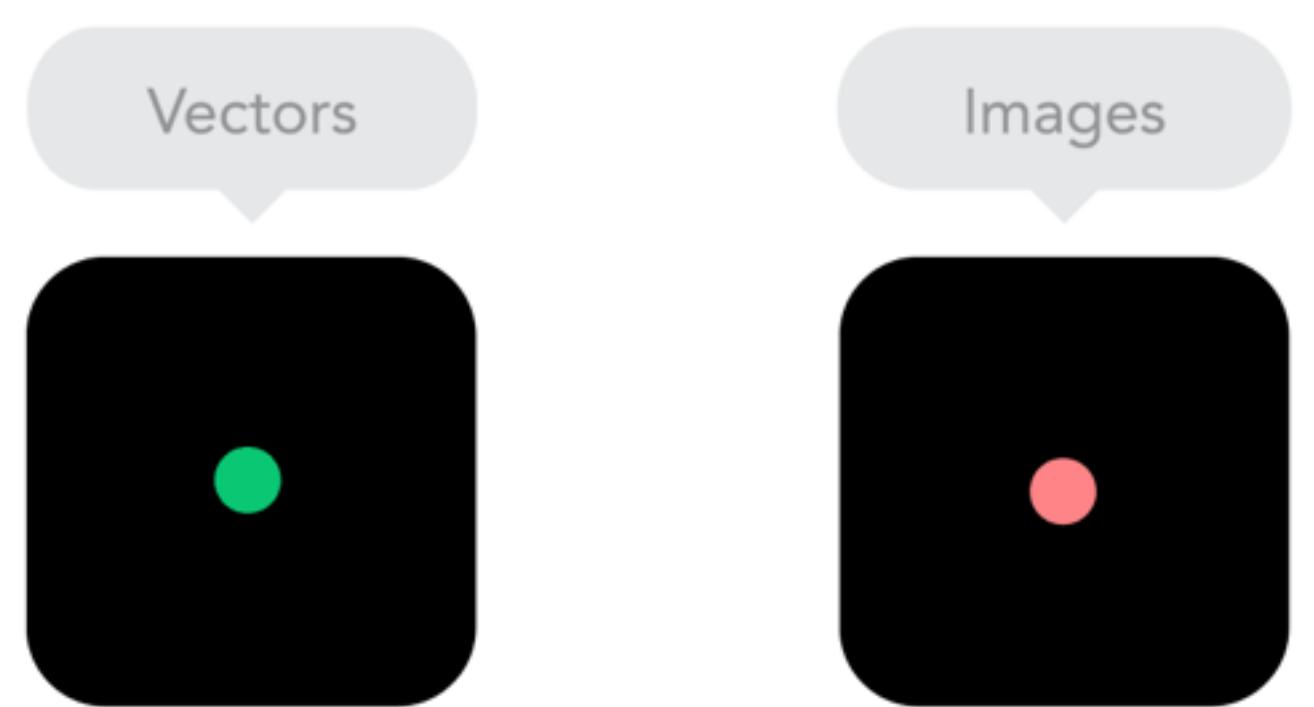


Webfonts



# 벡터 VS 비트맵 / 그 래 / 피 /

기술을 이용해, 동료를 이용해, 세상을 이용해



# /TERM/

기술을 이용해, 동료를 이용해, 세상을 이용해

RWD

AWD

Flexible

Fluid(Liquid)

Static

Device Pixel Density

CSS Pixel

Breakpoint

Media Queries

pt

dp

px

em

rem

Vector (SVG)

Bitmap (PNG/JPG/GIF)

Layout

Navigation

Content

Context

Grid System

Row

Column

Gutter

Margin

Toggle

Stack

Progressive Enhancement

Graceful degradation

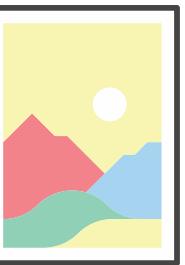
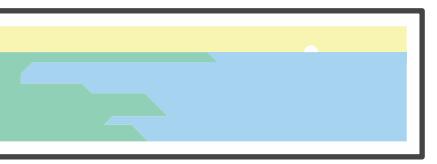
Mobile First

Modal

Caraousel

Responsive Web Design  
**RWD? AW?**

Adapt Web





### [기고] 반응형 웹의 진실: 트렌드 따라가다 내실 놓친다

[컴퓨터월드] '반응형 웹', IT트렌드라 불러다오!반응형 웹은 어떻게 트렌드가 됐나국내에 스마트폰이 도입된 지 불과 7년. 셀 수 없이 등장한 다양한 디바이스들은 시공간 제약을 무의미하게 만든 채 언제 어디서나 정...

DATE 2016.05.02 / AM 6:23

---

### [기고] 적응형 웹 전성시대: 반응형 웹의 대안이 될 수 있나

[컴퓨터월드] [그림1]은 최근 고객과의 기술미팅에서 빈번하게 발생하는 대화 내용이다. 나는 고객에게 명확한 답을 내려주지 않고 다시 질문한다. 어떤 시스템을 구현하고자 반응형 웹을 고려하고 있는지 말이다. 왜냐하면...

DATE 2016.06.01 / AM 11:06

---

### [기고] 반응형 웹과 적응형 웹을 넘어 경험을 디자인하다

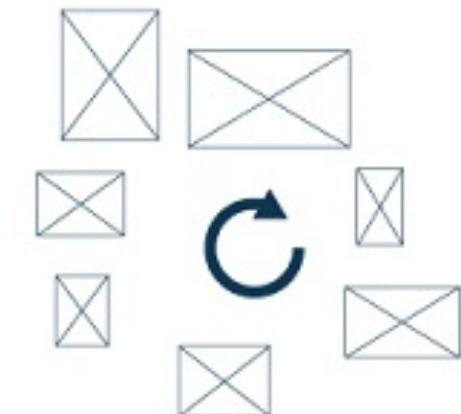
[컴퓨터월드] “우리 IT업계의 많은 사람들은 다양한 경험을 갖지 못했어요. 그래서 그들은 각각의 요소를 연관 지을만한 충분한 ‘연결고리’가 없었고, 문제에 대해 폭넓은 관점을 갖지 못한 채 매우 단선적인 솔루션을...

DATE 2016.07.01 / AM 9:26

반응형 웹 장점



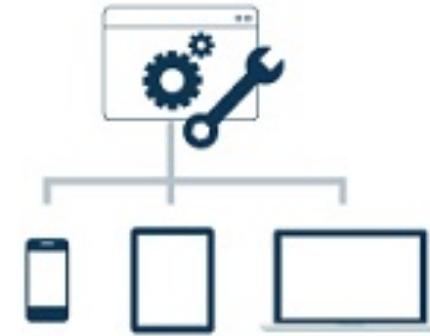
기기 화면 크기/해상도에 구애 받지 않고  
웹 사이트 레이아웃 구성



화면 변화, 웹 브라우저 사이즈 변화에  
즉시 반응



어느 기기에서든 사이트 주소 동일



기기마다 따로 코드 관리 필요 없어  
유지 보수 편리



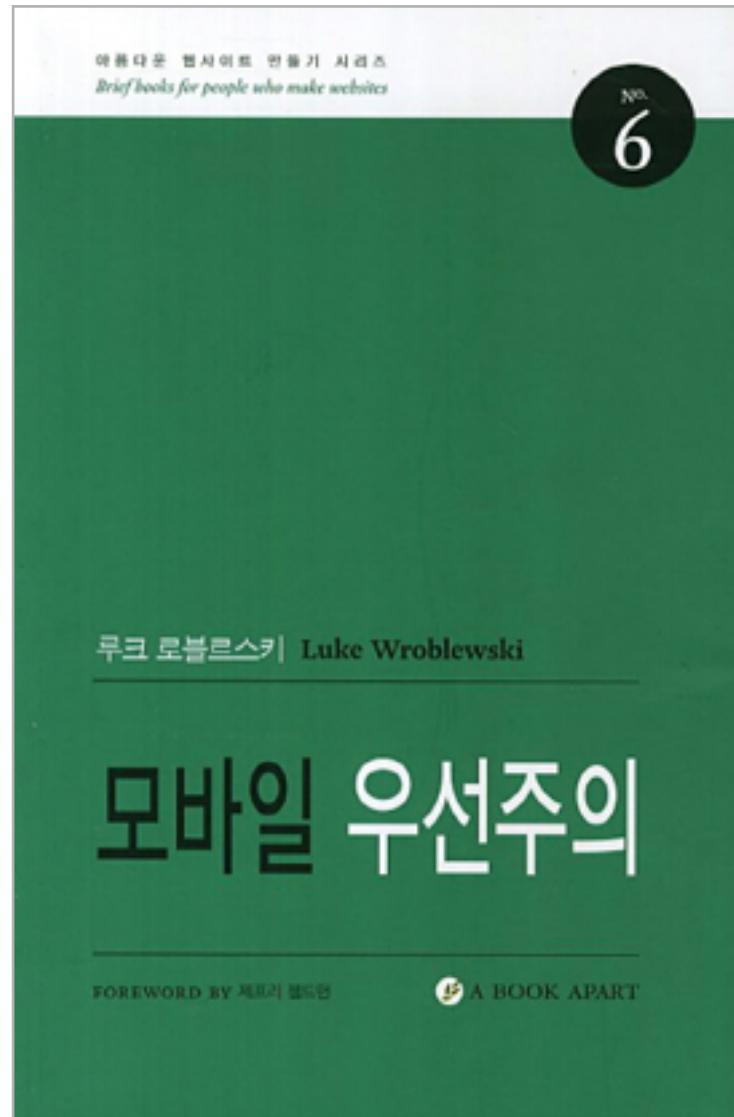
국제 표준인 웹 표준 기술로 구현

반응형 웹의 지향점은 참으로 이상적인 듯하나, 단순히 **해상도와 화면 사이즈를 조절해 많은 정보를 압축해 한 화면에 담아내는 것이 사용자에게 ‘진정한 만족감’을 줄 수 있을지는 생각해봐야 할 문제다.**

## 모바일 우선



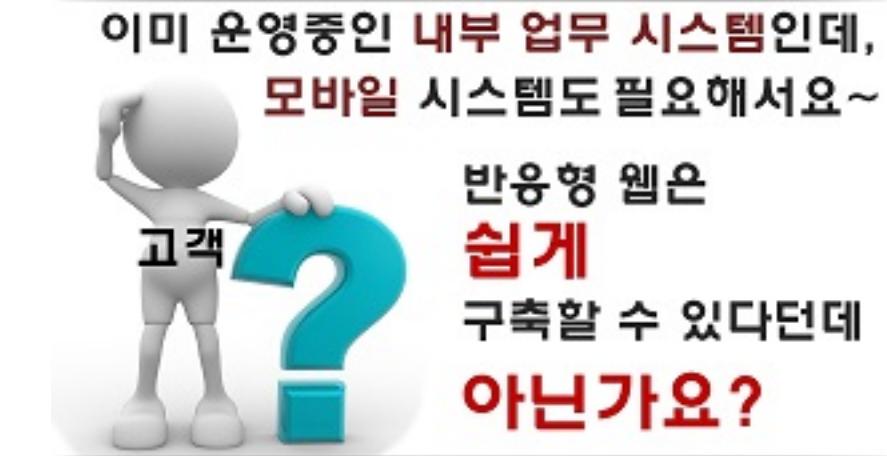
기술을 이용해, 동료를 이용해, 세상을 이용해  
이 전략은 우리나라 IT시장에서 난항을 겪을 수밖에 없다. **우리는 많은 정보를 압축해**  
**한 화면에 표현하는 것을 선호하는 편이다.** 이는 국내기업이 운영하는 대표 포털사이트  
'네이버'와 미국 기업이 운영하는 포털사이트 '구글' 첫 페이지지만 비교해봐도 쉽게  
이해할 수 있을 것이다.



기술을 이용해, 동료를 이용해, 세상을 이용해

대다수 모바일 UI 전문가들은 책에서 다루고 있는 내용이 대부분 꽤 익숙할 것이다. 이미 자신들도 많이 고민했던 문제 이거나 해법도 대략 알고 있는 문제일 것이다. 다만 내 입장에서 한 가지 느낀 점이 있다면 2003-4년부터 모바일 프로젝트를 진행하면서 **모바일의 작은 화면이 제약이라고만 생각했지 기회라고 생각하지 않았다는 것이다.**

이제와서 생각해 보면, 저자의 말대로 우리는 PC의 화면이 너무 넓어서 쓸데없는 것들을 채워 넣지는 않았나 반성해 보게 된다. 모바일이 나왔을 때, 진짜 UI 전문가라면 '그래 이거야. 마침내 사용자가 자신의 태스크에 집중 할 수 있는 환경이 나타났군'이라고 생각해야 하는데, **기존 매체에 대한 고정관념에 사로 잡혀 화면이 작다는 것은 제약이라고 생각했던 것 같다. 앞으로 또 다른 매체가 나타나고 다른 사람들이 입을 모아 '제약'이라고 말하는 것들을 기회로 바꿔 볼 생각이다.**



최근 고객과의 기술미팅에서 빈번하게 발생하는 대화 내용이다. 나는 고객에게 명확한 답을 내려주지 않고 다시 질문한다. 어떤 시스템을 구현하고자 반응형 웹을 고려하고 있는지 말이다.

왜냐하면 고객이 생각하는 이상적인 부분과 현실 사이에는 간극이 제법 크기 때문이다. 고객 입장에선 최근의 웹표준 기술을 감안할 때 ‘CSS 미디어 쿼리 좀 쓰면 되는 거 아닌가?’라고 쉽게 생각할 수 있는데, 실제로는 시스템 성격에 따라 검토해야 할 사항이 많다.

# NAVER

<http://www.naver.com>



<http://m.naver.com>



모바일

# Daum

<http://www.daum.net>



<http://m.daum.net>



모바일

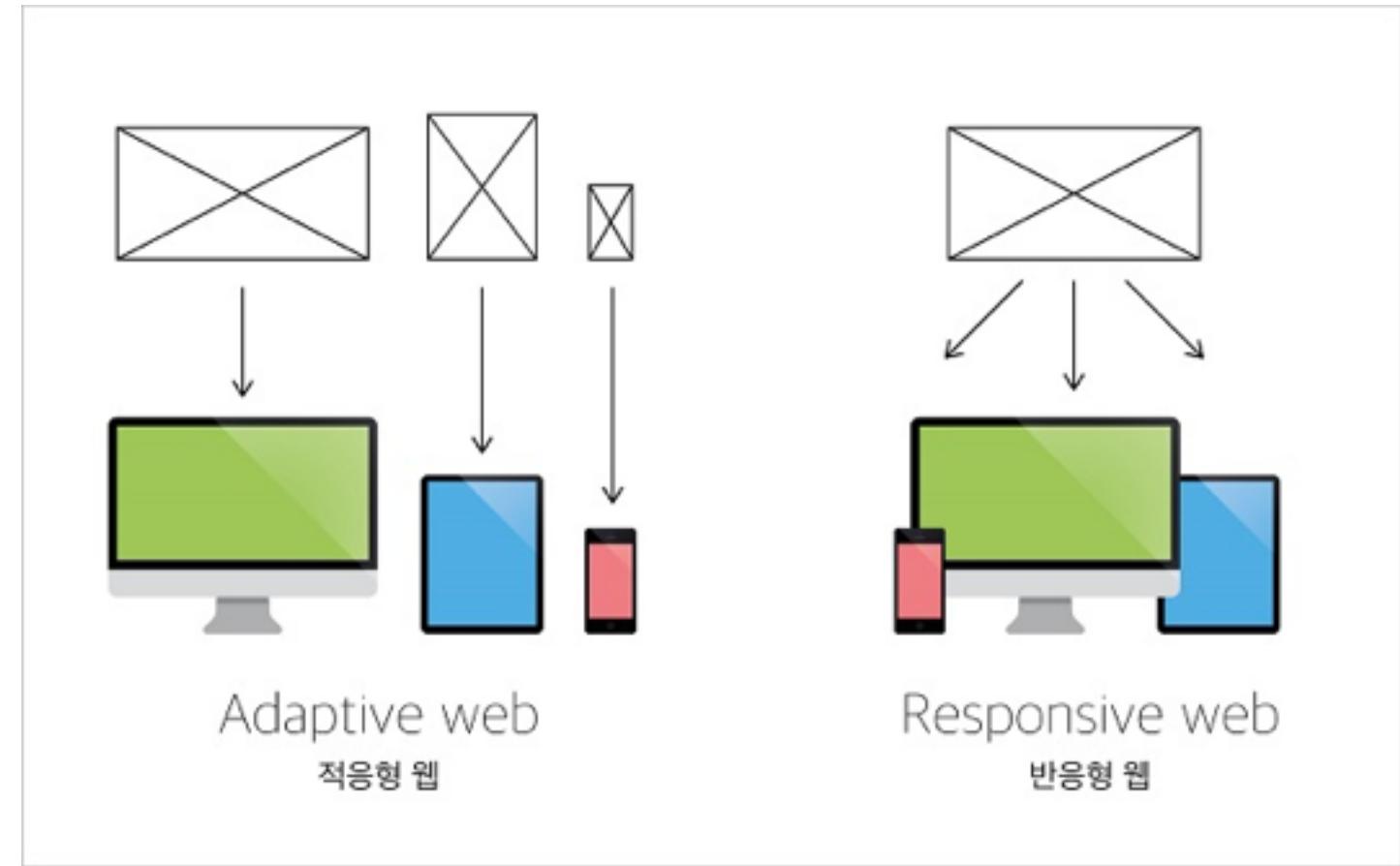
기술을 이용해, 동료를 이용해, 세상을 이용해

일반 홈페이지의 경우 기업 업무시스템이나 쇼핑몰·포털사이트 등에 비해 포함하고 있는 콘텐츠 양이 적은 편이라, 전달하려는 내용을 한 페이지에 표현하기 수월하다. 따라서 해상도 변경에 따라 화면의 레이아웃을 조절하는 반응형 웹을 적용하기 어렵지 않다.

반면, 앞서 말한 복잡한 기업 업무시스템이나 콘텐츠 내용이 많은 쇼핑몰, 포털 사이트 등은 어떻게 구현할 것인가? 국내 대표적인 포털사이트 ‘네이버’와 ‘다음’을 살펴보자. 두 포털사이트는 PC와 모바일에서의 접속 URL이 다르다. 이는 PC와 모바일 HTML 화면 자체가 다르다고 보면 된다.



기술을 이용해, 동료를 이용해, 세상을 이용해



반응형 웹은 한번의 개발로 디스플레이 종류에 따라 화면 크기 및 해상도가 자동으로 조절돼 화면을 보여주는 방식인데 비해, 적응형 웹은 미리 정해진 몇 가지 화면 크기를 기준으로 두고 비율에 맞춰 페이지를 구성하는 방식이다.

‘적응형 웹’이라는 용어는 Aaron Gustofson이 2011년 출간한 그의 책 ‘Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement’를 통해 처음 알려졌다. 그는 반응형 웹 개념을 포함하는 포괄적인 개념으로 적응형 웹을 설명했으나, 국내에서는 두 개념의 기술적인 부분에 초점이 맞춰져 통용되고 있다.

반응형 웹 디자인을 보완하는 적응형 웹의 특징

# 1. 콘텐츠의 양이 많을 경우

기술을 이용해, 동료를 이용해, 세상을 이용해

국내 시스템들은 많은 정보를 압축해 한 화면에 표현하는 것을 선호한다. 단순한 웹페이지나 홈페이지의 경우는 그나마 콘텐츠가 단순하나, 소셜커머스나 기업 업무시스템들은 콘텐츠의 양이 많고 복잡한 화면들이 대부분이다. 따라서 미디어 쿼리 등을 이용한 단순 레이아웃 변경 방식인 반응형 웹으로 콘텐츠의 배치와 배열을 정리하기란 쉽지 않은 일이다. 반면, **적응형 웹은 기획 첫 단계부터 기기에 알맞은 해상도를 고려한다. 화면에 담길 콘텐츠의 양을 조절하고 기기에 최적화된 디자인을 함으로써 가독성을 높이는데 상대적으로 자유롭다.**

반응형 웹 디자인을 보완하는 적응형 웹의 특징

## 2. 빈번한 그리드 형태 데이터 표현

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

반응형 웹은 그리드(Grid) 형태로 데이터를 표현하는 것이 쉽지 않다. 행(Row)과 열(Column)을 표현하는 그리드에서 행의 개수는 비교적 익숙하게 사용하는 상하 스크롤로 표현한다지만, 해상도에 따라 열의 데이터가 안 보이는 경우도 발생하고 가로축으로도 스크롤이 생겨서 정보를 확인하는데 불편함을 야기한다. 하지만 **적응형 웹은 디바이스 화면 사이즈에 맞는 열의 개수를 감안해 디자인할 수 있으므로 상하좌우로 스크롤바가 생기는 불편함을 없앨 수 있다.**

반응형 웹 디자인을 보완하는 적응형 웹의 특징

### 3. 시스템 성능이 중요한 경우

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

반응형 웹은 기기별 CSS스타일과 개발 소스들을 전부 로딩하기 때문에 시스템 로딩 속도나 성능에 영향을 주는데 비해, 적응형 웹은 제작한 기기에 해당하는 CSS스타일과 개발 소스만 로드 하기 때문에 상대적으로 성능 이슈가 적다.

이처럼 적응형 웹은 기기별 맞춤형 기획을 진행하고 및 개발하는 방식을 택해 반응형 웹의 아쉬운 부분들을 보완한다. 이에 앞서 언급한 대형 포털사이트 네이버나 다음 역시 PC와 모바일 페이지의 URL 주소를 별개로 두고(즉, 기기별 페이지를 분리함으로써) 기기에 따라 높은 만족도의 사용자 경험을 제공하고 있는 것이다.

### 적응형 웹 적용이 필요한 경우

## 1. 오래 전 구축된 레거시 시스템

레거시 시스템(Legacy System)은 현재까지 쓰이고 있거나 영향을 주고 있는 낡은 기술이나 방법론, 컴퓨터 시스템, 소프트웨어 등을 말한다. 기업의 레거시 시스템들은 짧게는 수년간, 길게는 몇십 년간 사용돼온 PC 위주 시스템으로, PC에 최적화된 디자인과 기능들로 구성돼있다.

따라서 단순하게 콘텐츠의 배치와 크기 조절을 통해 모바일 시스템을 구축하는 것은 현실적으로 어렵다. 이 때문에 레거시 시스템을 활용하고 있는 기업에서는 적응형 웹 구현방식으로 새롭게 모바일 시스템을 구축하는 경향이 있다.

적응형 웹 적용이 필요한 경우

## 2. 특수한 목적성을 띤 시스템

일반적인 홈페이지나 이벤트성 페이지 등 대고객 서비스를 지향하는 웹은 처음부터 반응형 웹을 검토해 적용하는 것이 편할 수 있다. 하지만 예를 들어 택배, A/S 기사처럼 특정 목적을 위해 시스템을 사용하는 사용자들은 본인이 활용하는 디바이스에 최적화된 시스템을 사용하길 원한다.

따라서 이들이 효율적으로 시스템을 활용할 수 있도록 적응형 웹을 활용해 디바이스에 최적화된 시스템을 구현하는 것이 좋다.

### 적응형 웹 적용이 필요한 경우

## 3. 유지보수 편의를 우선시하는 시스템

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

앞서 다룬 두 가지 경우에 포함되지는 않으나 여러 가지 이유로 반응형 웹을 적용하기 난해한 시스템이 있을 수 있다. 이런 경우 애초에 다른 환경에서 구동되는 시스템을 각각 분리해 기획·개발하는 것이 효율적일 수 있다. 한 번의 개발 또는 하나의 소스로 모든 환경에 알맞은 시스템을 설계하고 구현하는 것이 이상적 이긴 하나, 그 실현을 위한 고려사항이 생각보다 많고 원하는 성능을 낼 수 있을지도 장담할 수 없기 때문이다. 그래서 처음부터 각 사용자 환경을 별도로 분리해 처리하는 것이다.

또한 기간계 업무시스템처럼 상대적으로 오랜 기간 동안 유지보수를 지속적으로 진행하는 시스템은 기기에 따라 비즈니스 로직이 달라지고, 그에 따라 화면이 달라지기도 한다. 그러므로 적응형 웹을 고려하는 것이 현명할 수 있다.

### 적응형 웹 적용이 필요한 경우

## 3. 유지보수 편의를 우선시하는 시스템

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

앞서 다룬 두 가지 경우에 포함되지는 않으나 여러 가지 이유로 반응형 웹을 적용하기 난해한 시스템이 있을 수 있다. 이런 경우 애초에 다른 환경에서 구동되는 시스템을 각각 분리해 기획·개발하는 것이 효율적일 수 있다. 한 번의 개발 또는 하나의 소스로 모든 환경에 알맞은 시스템을 설계하고 구현하는 것이 이상적 이긴 하나, 그 실현을 위한 고려사항이 생각보다 많고 원하는 성능을 낼 수 있을지도 장담할 수 없기 때문이다. 그래서 처음부터 각 사용자 환경을 별도로 분리해 처리하는 것이다.

또한 기간계 업무시스템처럼 상대적으로 오랜 기간 동안 유지보수를 지속적으로 진행하는 시스템은 기기에 따라 비즈니스 로직이 달라지고, 그에 따라 화면이 달라지기도 한다. 그러므로 적응형 웹을 고려하는 것이 현명할 수 있다.

## 적응형 웹의 극복 문제

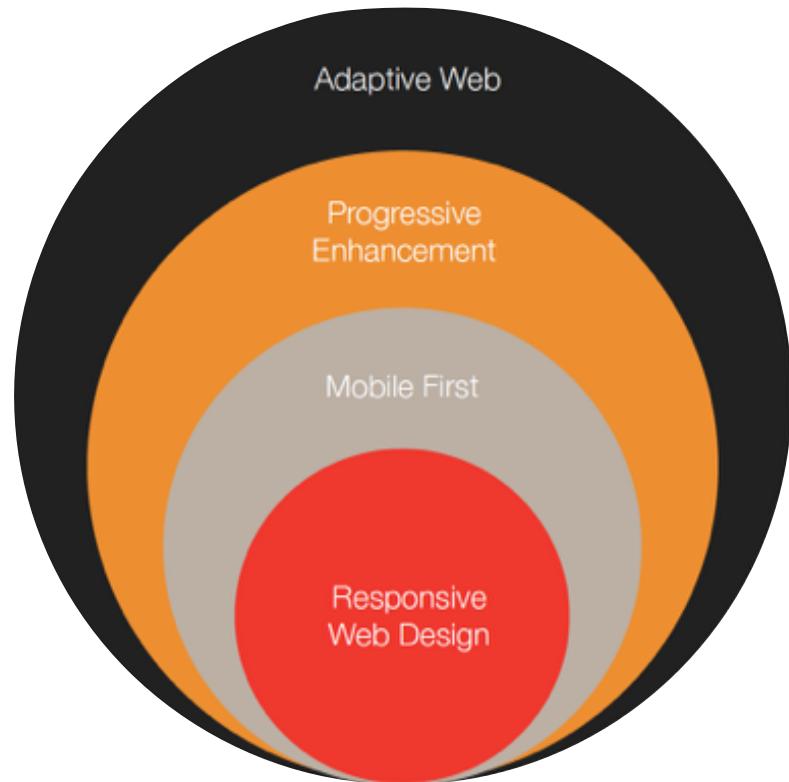
# 적응형 웹이 가진 문제점

기술을 이용해, 동료를 이용해, 세상을 이용해

IROPKE Front-End Design CAMPUS

**웹 분리 개발에 따른 비용 문제가 있을 수 있다.** 결국은 디바이스에 적절한 페이지 혹은 시스템을 각각 제작해야 하므로, 그 비용에 대한 효용성을 고려해야 하는 것이다. 또 **분리해 개발한 웹에 수정사항이 발생하면 다른 환경에 최적화된 나머지 웹페이지도 수정·변경해야** 하므로 개발자로 서는 다소 번거로워 질 수 있다. **새로운 디바이스가 시장에 나오면 이에 대한 고려도 해야 하는데,** 특히 대고객 서비스의 경우 검색엔진최적화(SEO) 측면도 신경 쓰이는 부분일 수 있다.

여전히 반응형 웹이 N스크린(N-Screen) 대응을 위한 대표적인 방법론으로 거론되고 있는 것은 적응형 웹이 성공사례를 축적해나가며 극복해야 할 중요한 과제 중 하나다. 적응형 웹의 성공사례를 통해 사용자경험을 고려하지 않은 채 구현된 반응형 웹이 ‘N스크린이 양산해내는 문제점들을 근본적으로 해결했다고 봐야 하는 것인지’에 대해 지속적으로 의문을 제기하고, 많은 사람들의 인식 속에 뿌리내린 ‘반응형 웹에 대한 환상들’을 바로잡아 나가야 할 것이다.



기술을 이용해, 동료를 이용해, 세상을 이용해

## 반응형 웹을 포함하는 적응형 웹

반응형 웹 디자인과 적응형 웹을 이론적으로 접근하는 것이 아닌, 경험 적이고 실제적 으로 접근해야 함을 강조하고 싶다. **반응형 웹과 적응형 웹을 상황에 맞게 적절하게 사용하는 것이 ‘N스크린 대응과 사용성’이라는 두 마리 토끼를 잡는 방법이 될 것**이며, ‘PC부터 모바일에 이르기까지 얼마나 만족할만한 사용성을 실제로 제공할 수 있는가’에 대한 고민이 우선시 돼야 한다는 것이다.

## “결국은 사람이다.”

스티브 잡스가 살아생전에 했던 말이다. 스티브 잡스는 늘 사용자를 지향하는 기술을 실현하고자 했고, 이에 관련된 다양한 시도와 실패를 되풀이 하곤 했다. 그는 다른 사람들이 깨닫기 이전에 UX의 중요성을 인지 했고, 그 결과 전 세계적으로 큰 영향을 미친 위인 중 한 명으로 거론 되기에 이르렀다.

그가 집요할 정도로 고집스럽게 집중했던 것은 제품과 서비스 자체가 아니라 ‘사람’이었다. 만약 스티브 잡스에게 “반응형 웹이 나을까요? 적응형 웹이 나을까요?”라고 묻는다면, 아마도 다음과 같이 대답하지 않을까?

**“그 어떤 웹이든, 자연스러운 인간의 행동을 유도하는 웹이 좋겠습니다.”**

물론 구체적으로 적용하는 기술과 기법들도 중요하지만, ‘사용자’에 대한 배려는 절대 간과해서는 안 되는 필수요소다. **사용자들이 어떻게 하면 시스템이나 웹페이지를 편하고 유용하게 사용할 수 있을지 근본적인 방향으로 고민하고 검토 하려는 노력이 필요하다.**



기술을 이용해, 동료를 이용해, 세상을 이용해

/감사합니다/