FEWD

GA GENERAL ASSEMBLY

EASY LAYOUT

WITH GUY ROUTLEDGE

@GUYROUTLEDGE | #FEWD

OBJECTIVE

Add more layout tools to our repetoir

WE CAN BUILD STUFF LIKE THIS...

BUT WHAT ABOUT THIS?

http://codepen.io/zastrow/full/kLeDa

AGENDA

- Resets and Boilerplates
- Building with Grids
- CSS Positioning

CSS RESETS

CSS RESET

There are a lot of annoying default styles that are inconsistent across browsers. Sometimes we might want to remove them all so we have a blank canvas.

SIMPLE RESET

```
* {
    margin: 0;
    padding: 0;
}
```

ERIC MEYER'S RESET

One approach:
http://meyerweb.com/eric/tools/css/reset/

NORMALIZE

My preference: http://nicolasgallagher.com/about-normalize-css/

STARTER TEMPLATE

I've created a starter template for your future projects.

It includes the valid HTML5 document setup, a CSS reset, a CSS file for your own styles, and even some Javascript stuff (more on that in a few weeks).

HTML5 BOILERPLATE

For a more advanced starter template, check out the HTML5 Boilerplate project.

http://www.h5bp.com

CSS FLOAT: PART TWO

FLOATS ARE WEIRD

http://codepen.io/guyroutledge/pen/oDwye

CLEARFIX

When floating elements, the parent container will collpase in height if it only contains floating elements.

This is because the floating elements have floated out of the container and the container doesn't know how to correctly calculate the height.

CLEARFIX

To work around this, we use the "clearfix" technique which is like putting a wedge in the top of the container and a wedge in the bottom of the container and forcing it open so it can correctly calculate the height.

CLEARFIX

Add this snippet to your CSS so you can clearfix collapsing containers

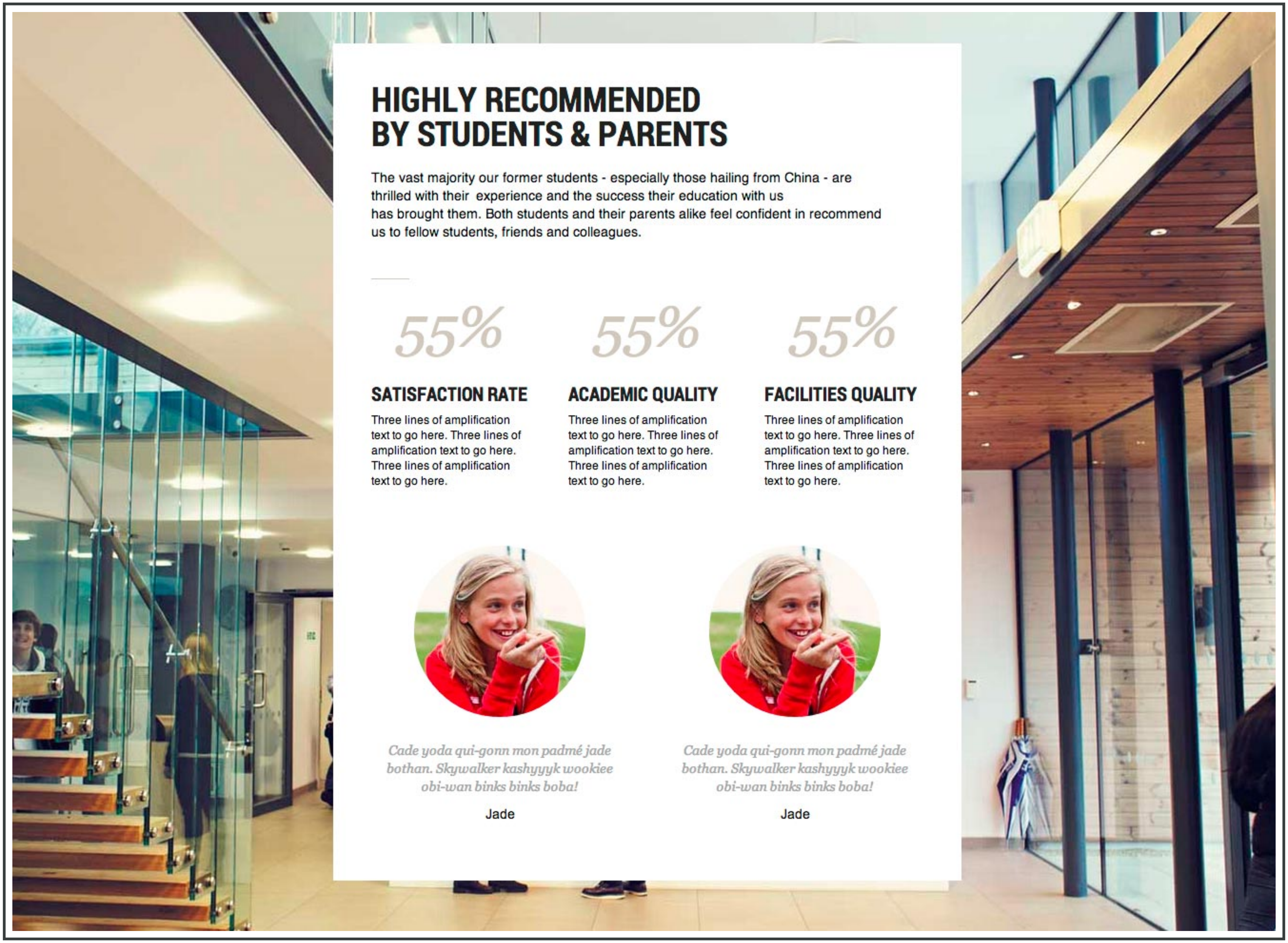
```
.clearfix:before,
.clearfix:after {
    content:" ";
    display:table;
}
.clearfix:after {
    clear:both;
}
```

CLEARFIX

To prevent a container from collapsing we add the **`clearfix`** class to the container in the HTML

```
<div class="main-content clearfix">

</div>
```



HIGHLY RECOMMENDED BY STUDENTS & PARENTS

The vast majority our former students - especially those hailing from China - are thrilled with their experience and the success their education with us has brought them. Both students and their parents alike feel confident in recommend us to fellow students, friends and colleagues.

55%

SATISFACTION RATE

Three lines of amplification text to go here. Three lines of amplification text to go here. Three lines of amplification text to go here.

55%

ACADEMIC QUALITY

Three lines of amplification text to go here. Three lines of amplification text to go here. Three lines of amplification text to go here.

55%

FACILITIES QUALITY

Three lines of amplification text to go here. Three lines of amplification text to go here. Three lines of amplification text to go here.

Cade yoda qui-gonn mon padmé jade bothan. Skywalker kashyyyk wookiee obi-wan binks binks boba!

Jade

Cade yoda qui-gonn mon padmé jade bothan. Skywalker kashyyyk wookiee obi-wan binks binks boba!

Jade

GRIDS

Imagine we have lots of pages that all use a combination of 1, 2, 3 and 4 columns for the layout

GRIDS

Could we use someting in CSS to standardise this?

GRIDS

Yes. `class` selectors are perfect for building grids as they can be repeated on many different elements.

We write a very small amount of CSS which is then reused many times throughout a page and throughout a project.

GRIDS

A really simple grid system

```
.grid-row { margin:0 -20px; }
.grid-column {
    float:left;
    margin:0;
    padding:0 20px; /* match to .grid-container */
}
.grid-quarter        { width:25%; }
.grid-third          { width:33.333%; }
.grid-half           { width:50%; }
.grid-two-third      { width:66.667%; }
.grid-three-quarter  { width:75%; }
.grid-whole          { width:100%; }
```

</>
CODE ALONG

USING A GRID

http://codepen.io/guyroutledge/pen/YqYoaW

CSS POSITIONING

CSS POSITIONING

- Precise placement of elements
- Tweak position of elements
- Not for layout
- Mostly for tackling complex design

CSS POSITION PROPERTY

CSS `position` can take 1 of 4 values.

`static` is the default and does not have to be set

- `static`
- `relative`
- `absolute`
- `fixed`

POSITION:RELATIVE

- Move elements **relative** to themselves
- Create a parent container
- Position elements absolutely inside

POSITION:ABSOLUTE

- Set the **absolute** co-ordinates of an element to be positioned
- `top`, `right`, `bottom`, `left`
- Use `px`, `%`, `em` and any other CSS unit

POSITION:FIXED

- **Fix** an element in place, regardless of scroll
- Set co-ordinates of where to fix an element
- Position is set relative to the **`body`**

Z-INDEX

- `z-index` controls the stacking order
- The z-axis comes out of the screen towards you
- It’s a bit like Photoshop layers
- `z-index` only works for `position` other than static

</> CODE ALONG

CSS POSITION

http://codepen.io/guyroutledge/pen/JGjLVL