Elements & Typography

Grid (Columns)

For laying out content on a page, the core hub framework includes styles for a 12-column grid.

...  
...  
...  
...  
...  
...  
...  
...  
...  
...  

The grid supports up to 12 columns with span# and offset# classes.

Each column must have a .col class. The last column in a set must have the .omega class added for IE 7 to work properly. No clearing div is required.

For example, a four column grid would look like:

```html
<div class="grid">
  <div class="col span3">
    ...
  </div>
  <div class="col span3">
    ...
  </div>
  <div class="col span3">
    ...
  </div>
  <div class="col span3">
    ...
  </div>
</div>
```
Spanning Columns

Columns can be spanned to easier portion content on the page. In the following example, we span the first 6 columns in a container, then follow with two, smaller 3 column containers for a 3-column layout where the first column takes up 50% of the space.

```html
<div class="grid">
  <div class="col span6">
    ...
  </div>
  <div class="col span3">
    ...
  </div>
</div>

Output:

```
Offsets

Columns may also be offset or 'pushed' over.

```html
<div class="grid">
  <div class="col span3 offset3">
    ...
  </div>
  <div class="col span3">
    ...
  </div>
  <div class="col span3 omega">
    ...
  </div>
</div>
```

Output:

```html
...
...
...
```

Helper Classes

- `.span-quarter` Span 3 columns. This is equivalent to `.span3`
- `.span-third` Span 4 columns. This is equivalent to `.span4`
- `.span-half` Span 6 columns. This is equivalent to `.span6`
- `.span-two-thirds` Span 8 columns. This is equivalent to `.span8`
.span-three-quarters
Span 9 columns. This is equivalent to .span9

A four column grid with the helper classes:

```html
<div class="grid">
  <div class="col span-quarter">
    ...
  </div>
  <div class="col span-quarter">
    ...
  </div>
  <div class="col span-quarter">
    ...
  </div>
  <div class="col span-quarter omega">
    ...
  </div>
</div>
```

There are equivalent .offset- classes as well:

- .offset-quarter
  Offset 3 columns. This is equivalent to .offset3
- .offset-third
  Offset 4 columns. This is equivalent to .offset4
- .offset-half
  Offset 6 columns. This is equivalent to .offset6
- .offset-two-thirds
  Offset 8 columns. This is equivalent to .offset8
- .offset-three-quarters
  Offset 9 columns. This is equivalent to .offset9

Markup for a four column grid with the offset helper class:

```html
<div class="grid">
  <div class="col span-quarter">
    ...
  </div>
  <div class="col offset-quarter span-quarter">
    ...
  </div>
  <div class="col span-quarter omega">
    ...
  </div>
</div>
```
Nesting Grids

The following is an example of a 3 column grid nested inside the first column of another 3 column grid.

```html
<div class="grid">
  <div class="col span6">
    <div class="grid">
      <div class="col span4">
        ...
      </div>
      <div class="col span4">
        ...
      </div>
      <div class="col span4 omega">
        ...
      </div>
    </div>
  </div>
  <div class="col span3">
    ...
  </div>
  <div class="col span3 omega">
    ...
  </div>
</div>
```

Notifications

The core framework provides some base styles for alter and notifications.

<p class="passed">Success message</p>

Success message

<p class="info">Info message</p>

Info message

<p class="help">Help message</p>

Help message

<p class="warning">Warning message</p>

Warning message
Sections & Asides

The majority of hub components have content laid out in a primary content column with secondary navigation or metadata in a smaller side column to the right. This is done by first wrapping the entire content in a div with a class of .section. The content intended for the side column is wrapped in a <div class="aside"> tag. The primary content is wrapped in a <div class="subject"> tag and immediately follows the .aside column.

Note: The .aside column must come first in order for the content to be positioned properly. If, unfortunately, this poses a semantic problem, we recommend using the grid system as a potential alternative.

Using aside & subject differs from the grid system in that the .aside column has a fixed width with the .subject column taking up the available left-over space. In the grid system, every column is flexible (uses a percentage of the screen) and cannot have a specified, fixed width.

Example usage:

```html
<section class="section">
  <div class="section-inner">
    <div class="aside">
      Side column content ...
    </div>
    <div class="subject">
      Primary content ...
    </div>
  </div>
</section>
```

Buttons

{xhub:include type="stylesheet" filename="/media/system/css/buttons.css"}

States
default disabled active

```html
<a class="btn" href="#">default</a>

<a class="btn disabled" href="#">disabled</a>

<a class="btn active" href="#">active</a>
```

Size

primary secondary

```html
<a class="btn btn-primary" href="#">primary</a>

<a class="btn btn-secondary" href="#">secondary</a>
```

Type

link button

```html
<a class="btn" href="#">link</a>

<button class="btn" href="#">button</button>

<input type="submit" class="btn" value="input" />
```

Color

danger warning info success

```html
<a class="btn btn-danger" href="#">danger</a>

<a class="btn btn-warning" href="#">warning</a>

<a class="btn btn-info" href="#">info</a>

<a class="btn btn-success" href="#">success</a>
```
Icons

danger
warning
info
success
edit
delete
<a class="btn btn-danger icon-danger" href="#">danger</a>
<a class="btn btn-warning icon-warning" href="#">warning</a>
...

Groups

Dropdown

- Action
- Another action
- Something else here
- Separated link

<div class="btn-group dropdown">
    <a class="btn" href="#">Dropdown</a>
    <span class="btn dropdown-toggle"></span>
    <ul class="dropdown-menu">
        <li><a href="#">Action</a></li>
    </ul>
</div>
Dropup

- Action
- Another action
- Something else here

- Separated link

Dropdown

- Action
- Another action
- Something else here

- Separated link

prev all next