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>
```

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 class="col span3 omega">...
  </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:

...
.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:

```html
.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 alert 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

<p class="error">Error message</p>

Error 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>
```

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

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

Size

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

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

Type

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

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

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

Color

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

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

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

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

danger

warning

info

success

edit

delete

delete

secondary

<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>
        9 / 11
    </ul>
</div>
<li><a href="#">Another action</a></li>
<li><a href="#">Something else here</a></li>
<li class="divider"></li>
<li><a href="#">Separated link</a></li>
</ul>
</div>

Dropup

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

<div class="btn-group dropup">
  ...
</div>

Dropdown

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

<div class="btn-group btn-secondary dropdown">
  ...
</div>