

# ON DISPLAY:

## A PRACTICAL OVERVIEW OF THE CSS DISPLAY PROPERTY

A presentation by **Steve Blaurock** / @sblaurock

# WHY DO WE USE THE *DISPLAY* PROPERTY?

1. To show and hide sets of elements.
2. To control the way elements "sit" in relation to one another.

# VALUES OF INTEREST

- CSS1
  - none
  - inline
  - block
  - list-item
- CSS2.1
  - inline-block
  - table
  - table-row
  - table-cell
- CSS3
  - flex
  - grid
  - Experimental
    - run-in

# DISPLAY: NONE (CSS1)

Visually remove an element (and its children) from the page.

**Element will have no effect on page layout.**

```
jQuery: $( '#elementId' ).hide( );
```

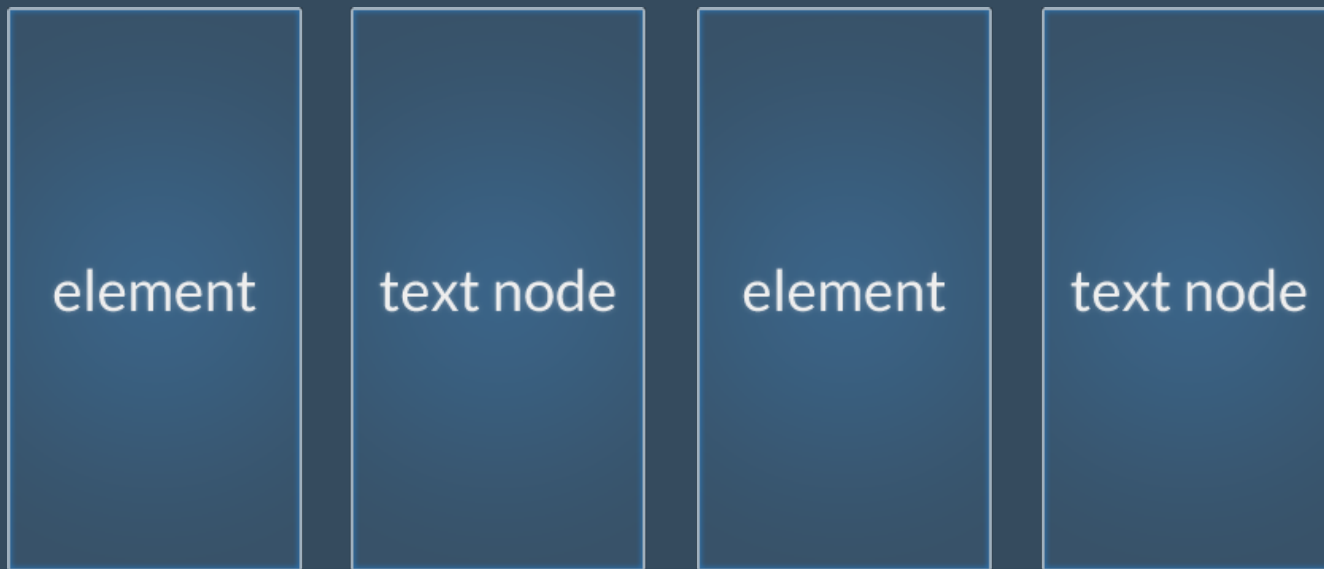
# VISIBILITY: HIDDEN (CSS2.1)

Render an element (and its children) as invisible.  
**Page layout will be effected by hidden element.**

[View Demonstration](#)

# DISPLAY: INLINE (CSS1)

`<span>`, `<a>`, `<b>`, `<img>` ...



Flow ↔

# DISPLAY: BLOCK (CSS1)

`<h1>, <p>, <div> ...`

element

element

element

element

Flow ⇅

# DISPLAY: INLINE-BLOCK (CSS2.1)

Follows natural inline flow, but gives us characteristics of block elements.

Lorem ipsum dolor sit 

I am text  
inside of an  
inline-block

 amet, consectetur  
adipiscing elit. Nulla id posuere neque, ut varius purus.

Flow ↔



# COMPARING WITH FLOAT

`inline-block` elements render the same way that text would.  
Adjust alignment using `vertical-align` if needed.

A square icon with a dark olive green background. Inside the square, the letters 'Aa' are written in a white, serif font. A thin white horizontal line is positioned directly beneath the lowercase 'a'.

[View Demonstration](#)

# ISSUES: SPACE IN MARKUP

Having spaces or newlines in your HTML markup will lead to unwanted space between elements.

```
<ul class="inline-block">  
  <li>Item A</li>  
  <li>Item B</li>  
  ...  
</ul>
```

[View Demonstration](#)

# ISSUES: IE7 COMPATABILITY

IE7 only allows `inline-block` on naturally inline elements.

```
.inline-block {  
  ...  
  /* IE7 Fix */  
  zoom: 1;  
  *display: inline;  
}
```

# DISPLAY: TABLE-\* (CSS2.1)

Allows for table characteristics to be applied through CSS.  
Useful for vertical centering.

```
<div style="display: table;">
  <div style="display: table-row;">
    <div style="display: table-cell;">
      YES, THIS IS CONTENT.
    </div>
  </div>
</div>
```

# OVERVIEW

## float

- + Full browser support.
- Parent container collapse (requires "clearfix hack").

## inline-block

- + No collapsing of parent container.
- + Full browser support(ish).
- Requires "hack" for IE7 support.
- Issue with whitespace in markup.

## table

- + No collapsing of parent container.
- + Allows for vertical centering.
- No support for IE7.
- Extra markup.

# DISPLAY: FLEX (CSS3)

The children of a “box” can be ordered either vertically or horizontally within a parent.

Flexbox allows you to control what happens to any remaining space. It is possible to nest these boxes, allowing for very complex layouts.

## Support:

Chrome (**Full**), Firefox (**Partial**), Safari (**Prefixed**),  
Opera (**Prefixed**), IE10 (**Partial**, **Prefixed**), IE11 (**Full**)

# IGNORED DISPLAY VALUES

`run-in` - Not supported in Firefox or IE7.

`grid` - Only supported in IE 10+ prefixed?

`list-item` - Just use real lists.

# SOURCES

CSS Tricks - Display

CSS Tricks - Fighting the Space Between Inline Block Elements

Design Shack - What's the Deal With Display: Inline-Block

MDN - Display

Mozilla Web Development - Cross Browser Inline-Block

Quirksmode - Display

# TOOLS

Hakim El Hattab / reveal.js



# THANK YOU!

Twitter: @sblaurock / Github: sblaurock