

12年3月21日星期三

2 hour

# Java Web - Front-end Basic

12年3月21日星期三

2 hour

# Java Web - Front-end Basic

Ji Hao

# Agenda

- HTML
  - basic, elements, attributes, link, form
  - doctype, head, meta, css, scripts, DOM
- CSS
  - basic, selectors, box, positioning
- JavaScript
  - objects, OO, DOM objects, jQuery

# HTML

# CSS - Selectors

- Basic

- Universal \*
- Type h1 h1,h2
- Class .class
- ID #id

- Hierarchy

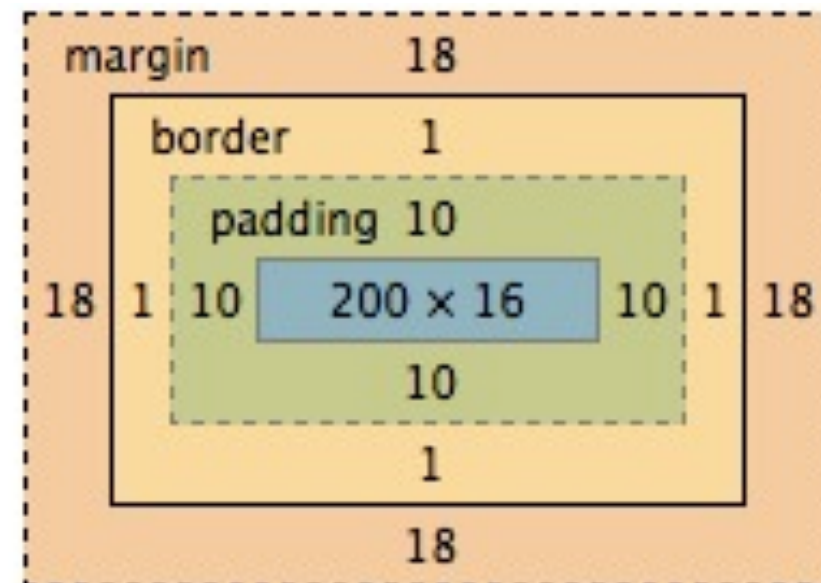
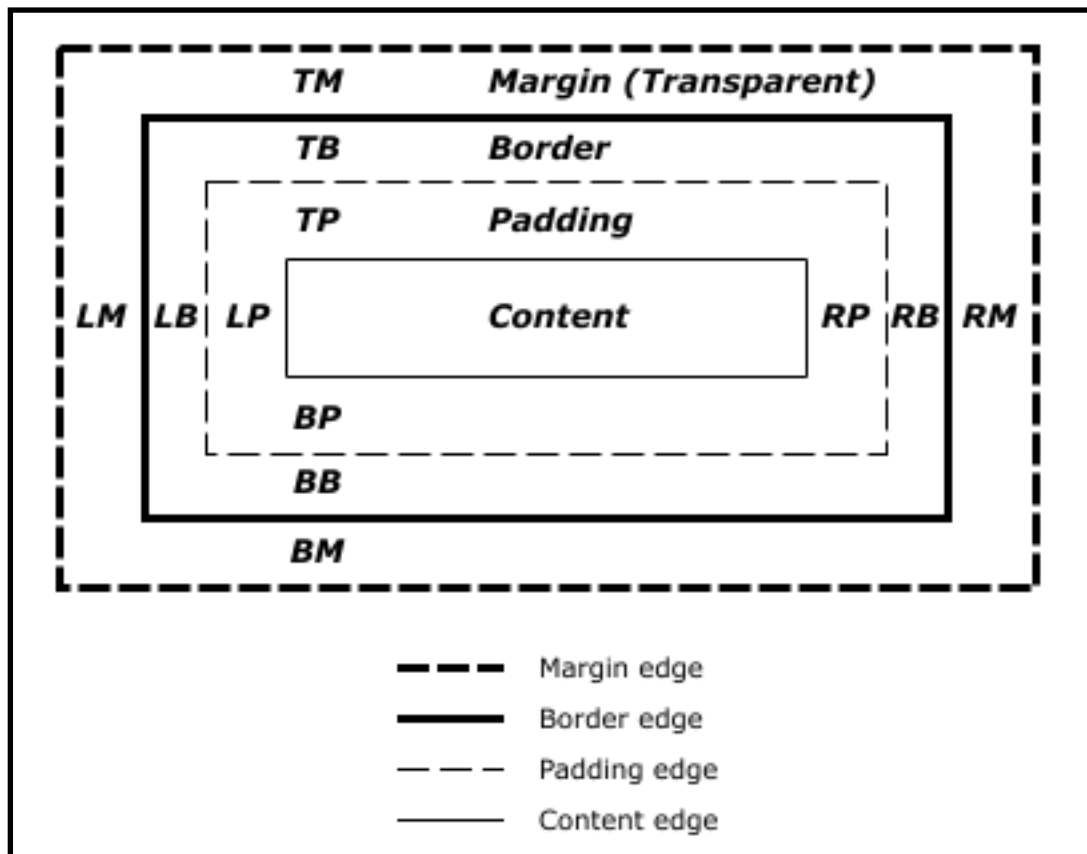
- Descendant div p div \* p
- Child div > p
- Adjacent sibling div + p div ~ p

- Filter

- Attribute h1[title] span[class="example"]
- Pseudo classes a:visited div > p:first-child

Live Demo

# CSS - Box model



```
body{  
    border:solid 1px #000;  
}  
#box{  
    width:200px;  
    border:solid 1px red;  
    padding: 10px 10px;  
    margin: 18px 18px;  
    background-color:#87C63A;  
}
```

<http://jsfiddle.net/jihao/kaLck/>

# CSS - Box model

**Margin properties:** 'margin-top', 'margin-right', 'margin-bottom', 'margin-left', and 'margin'

**Padding properties:** 'padding-top', 'padding-right', 'padding-bottom', 'padding-left', and 'padding'

## **Border properties**

Border width: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width', and 'border-width'

Border color: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', and 'border-color'

Border style: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style', and 'border-style'

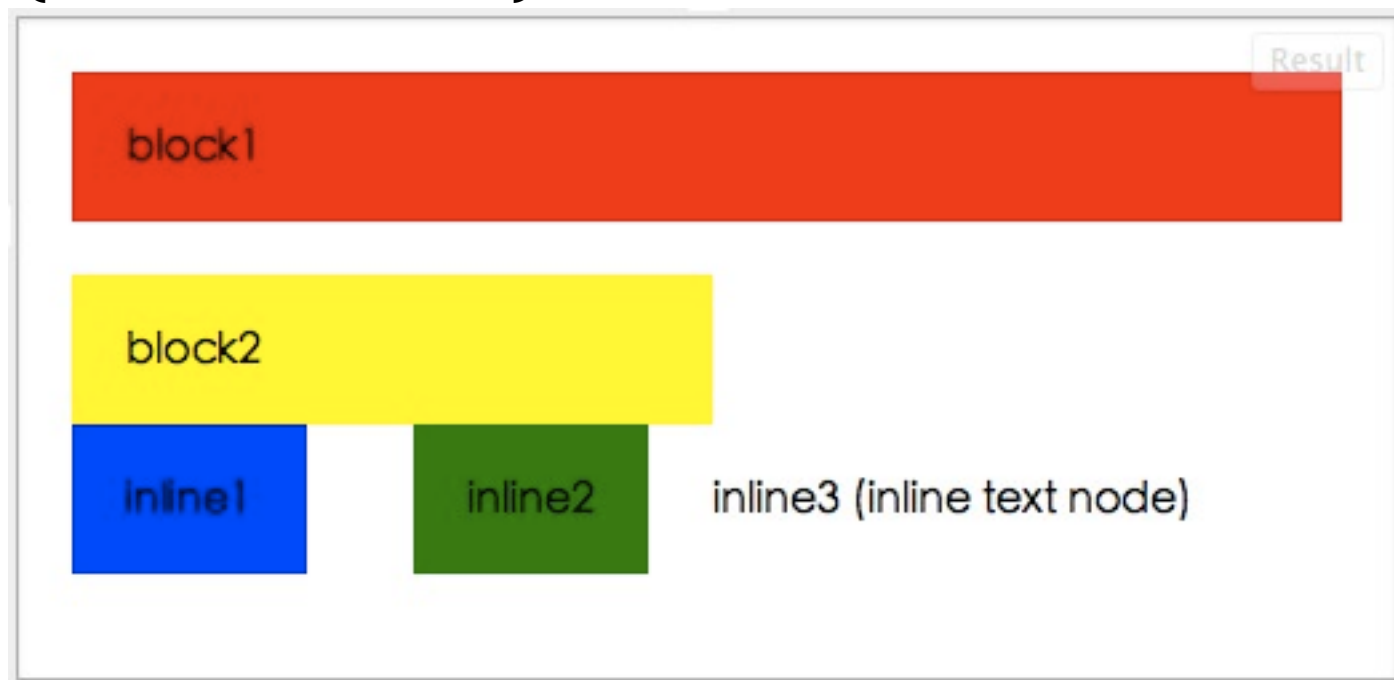
Border shorthand properties: 'border-top', 'border-right', 'border-bottom', 'border-left', and 'border'



# CSS - display: inline | block

```
p { background-color: red; padding: 20px; margin: 20px; }  
div { background-color: yellow; padding: 20px; margin: 20px; width: 200px; }  
span { background-color: blue; padding: 20px; margin: 20px; }  
strong { background-color: green; padding: 20px; margin: 20px; }
```

```
<p>block1</p>  
<div>block2</div>  
<span>inline1</span><strong>inline2</strong>  
<label>inline3</label> (inline text node)
```



<http://jsfiddle.net/jihao/jLM9z/>

# CSS - Positioning

- **Normal flow.** In CSS 2.1, normal flow includes **block formatting** of block-level boxes, **inline formatting** of inline-level boxes, and **relative positioning** of block-level and inline-level boxes.  
`position: static | relative`
- **Floats.** In the float model, a box is first laid out according to the normal flow, then taken out of the flow and shifted to the left or right as far as possible. Content may flow along the side of a float.  
`float: left | right`
- **Absolute positioning.** In the absolute positioning model, a box is removed from the normal flow entirely (it has no impact on later siblings) and assigned a position with respect to a containing block.  
`position: absolute | fixed`

# Normal flow

position: static | relative

- position:static;
  - HTML elements are positioned static by default.
  - Static positioned elements are not affected by the top, bottom, left, and right properties.
- position:relative;
  - A relative positioned element is positioned relative to its normal position.
  - The original position will still be occupied.

<http://jsfiddle.net/jihao/HCzFf/>

# Floats

float: left | right

- Elements are **floated horizontally**, this means that an element can only be floated left or right, not up or down.
- A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.
- The elements after the floating element will flow around it.
- The elements before the floating element will not be affected.
- use **clear** property to turn off floating

<http://jsfiddle.net/jihao/KWwyp/>

<http://jsfiddle.net/jihao/nV5T6/>

# Absolute positioning

- position: absolute
  - An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>
- position: fixed
  - An element with fixed position is positioned relative to the browser window.

<http://jsfiddle.net/jihao/A2a6f/>

# CSS - Positioning quiz

- How to implement this effect?
  - <http://twitter.github.com/bootstrap/components.html>

# CSS - Positioning quiz

- How to implement this effect?
  - <http://twitter.github.com/bootstrap/components.html>
- solution reference:

<http://jsfiddle.net/jihao/VYdhQ/>

# JavaScript

- ECMA-262
  - <http://www.ecmascript.org/>



# JavaScript - Objects

- Almost everything written in JavaScript is an object
- Objects can be thought of as a collection of properties —much like a hash in other languages
- JavaScript doesn't have a concept of classes like other object-oriented languages
- Classes are simulated using a concept called prototypal inheritance

# JavaScript - DataTypes

## A. Primitive:

- number - 1, 3, 1001, 11.12, 2e+3
- string - "a", "stoyan", "0"
- boolean - true | false
- null
- undefined

## B. Objects

- Object
- Number
- String
- RegExp
- Boolean
- Function
- everything else
- 

<http://jsfiddle.net/jihao/Cjh7v/>

# JavaScript - OO

// The old way

```
var myObject = new Object();  
myObject.val = "test";
```

// Using object literal notation

```
var myObject = {  
    val: "test"  
};
```

# JavaScript - OO

// The function + prototype way

```
var Book = function(name) {  
    this.className = "Book";  
    this.name = (name == undefined) ? "EMPTY" : name;  
};  
Book.prototype.getPrice = function() {  
    return this._price;  
};  
Book.prototype.setPrice = function(price) {  
    this._price = price;  
};  
  
var aBook = new Book("Advanced JavaScript");  
aBook.setPrice(50);  
var bBook = new Book();  
bBook.setPrice(60);  
document.write(aBook.name + " - " + aBook.getPrice());  
document.write(bBook.name + " - " + bBook.getPrice());
```

# JavaScript - OO

```
Book.prototype.getPrice = function() {  
    return this._price;  
};  
Book.prototype.setPrice = function(price) {  
    this._price = price;  
};
```

```
var aBook = new Book("Advanced JavaScript");  
aBook.setPrice(50);  
var bBook = new Book();  
bBook.setPrice(60);
```

```
//change the behavior of object bBook  
bBook.getPrice = function() {  
    return this._price + " $";  
};
```

```
document.write(aBook.name + " - " + aBook.getPrice());  
document.write(bBook.name + " - " + bBook.getPrice());
```

# JavaScript - OO

```
Book.prototype.getPrice = function() {  
    return this._price;  
};  
Book.prototype.setPrice = function(price) {  
    this._price = price;  
};
```

```
var aBook = new Book("Advanced JavaScript");  
aBook.setPrice(50);  
var bBook = new Book();  
bBook.setPrice(60);
```

```
//change the behavior of prototype Object  
Book.prototype.getPrice = function() {  
    return this._price + " RMB";  
};
```

```
document.write(aBook.name + " - " + aBook.getPrice());  
document.write(bBook.name + " - " + bBook.getPrice());
```

# Own properties vs. prototype's

```
Book.prototype.getPrice = function() {  
    return this._price;  
};  
Book.prototype.setPrice = function(price) {  
    this._price = price;  
};
```

```
var aBook = new Book("Advanced JavaScript");  
aBook.setPrice(50);  
var bBook = new Book();  
bBook.setPrice(60);
```

<http://jsfiddle.net/jihao/jkRnZ/>

```
console.log(aBook.hasOwnProperty('getPrice'));  
console.log(aBook.__proto__.hasOwnProperty('getPrice'));  
console.log(aBook.hasOwnProperty('name'));
```

```
console.log(bBook.hasOwnProperty('getPrice'));  
console.log(bBook.hasOwnProperty('name'));
```

# Browser Objects

- Window
- Screen
- History
- Location
- Navigator



# DOM Objects

- Live demo
  - a bookmarklet tool for quick access TR
  - open a new browser window
  - create a form on the fly
  - create event handling hooks on the fly

<http://jsfiddle.net/jihao/zQtpv/>

# JavaScript Libraries



# Assignment

- you will implement a login function
- submit a login form with get method, after form post display a greeting message for the user signed in
- set the login form size to 400px\*400px, place it to the horizontal center of screen, it shall work with different screen resolution
- use javascript to do input validation, display an error message if user didn't input username or password

# Q&A

# Tools

- Chrome + developer tools
- Firefox + firebug
- <http://www.my-debugbar.com/wiki/IETester/HomePage>
- <http://jsfiddle.net>

# References

- <http://w3schools.com>
- <http://www.w3school.com.cn>
- <http://www.w3.org/TR/CSS2>