

HTML

(Hyper Text Markup Language)

概要

- Web上の文章を記述するための言語
- 最新バージョンはHTML5

HTMLの基本

- タグ(要素)
文章の構造を指定
- 属性=属性値
タグで指定できない細かい内容を指定
- id
idを振り分けることで、識別を簡単に(固有)
- class
idに類似、複数に指定。

HTMLの基本



タグ

<html>

<body>

body要素

font要素

</body>

</html>

タグ:<要素名>～内容～</要素名> ※とりあえず

HTMLの基本



CSSで指定しよう！(後ほど)

属性=属性値

```
<html>
  <body>
    body要素 <BR>
    <font color="red" size="10">
      font要素 <br>
    </font>
  </body>
</html>
```

属性:<要素名 属性=属性名>～内容～</要素名>で終了

※タグは非推奨

HTMLの基本構造

<HTML>

<HEAD><TITLE>～</TITLE>
※主にメタ情報
</HEAD>

<BODY>

※主にWebページの内容

</BODY>

</HTML>

その他、タグなど(極一部)

タグ	使用例	内容
		画像の表示
<a>	リンク名	リンクの作成
<button>	<button type=“button”>ボタン名</button>	ボタンの作成
<table>	省略	表の作成
<pre>	<pre>内容</pre>	文章をそのまま反映
<p>	<p>文書</p>	段落の区別
<div>	<div>****</div>	ブロックの指定
	文章	ボールドで表示

CSS

(Cascading Style Sheets)

概要

- HTMLの属性値指定は面倒
- CSSを使うとその辺がかなり楽になる

CSSの基礎

タグ

属性

属性値

p{

font-size : 12pt ;

color : red

}

“;”で仕切り

CSSの基礎（HTML内）

Styleタグで指定

<style type="text/css">

<!--

Pタグ全てに反映

p { font-size : 10pt }

-->

</style>

CSSの基礎（HTML内）

idname を指定することで
反映させる範囲を指定できる

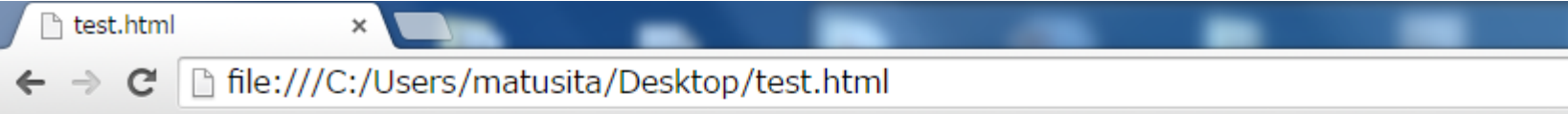
```
<style type="text/css">
```

```
<!--
```

```
#idname { font-size : 10pt }
```

```
-->
```

```
</style>
```



この文字のサイズは変わりません。

この部分の文字サイズは30ポイントになります。

```
<html>
<head>
  <style type="text/css">
    <!--
      p{ font-size : 30pt;
        color : red
      }
    -->
  </style>
</head>

<body>
  この文字のサイズは変わりません。
  <p>この部分の文字サイズは30ポイントになります。</p>
</body>
</html>
```

CSSの基礎（HTML内）

classname を指定することで
反映させる範囲を指定できる

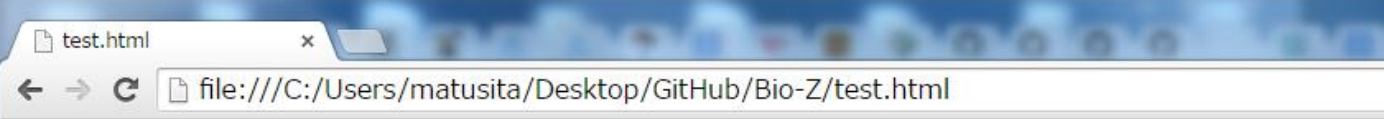
```
<style type="text/css">
```

```
<!--
```

```
.cname { font-size : 10pt }
```

```
-->
```

```
</style>
```



この文字のサイズは変わりません。

この部分の文字サイズは30ポイント になります。

この部分の文字サイズも30ポイント になります。

```
<html>
<head>
  <style type="text/css">
    <!--
      .cname{ font-size : 30pt;
        color : red
      }
    -->
  </style>
</head>
<body>
  <p>この文字のサイズは変わりません。</p>
  <p class = "cname">この部分の文字サイズは30ポイント
    になります。</p>
  <p class = "cname">この部分の文字サイズも30ポイント
    になります。</p>
</body>
</html>
```

CSS

path

リンク先が何か指定
ここではスタイルシート

<HEAD>

<LINK href="sample_style.css" rel="stylesheet"
type="text/css">

</HEAD>

HEADの中にリンク指定

Sample_style.cssの中は

この羅列

<style type="text/css">

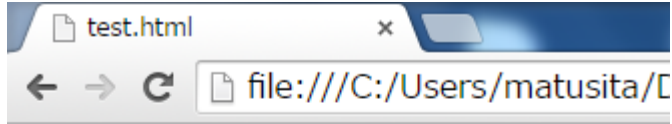
<!--

#idname { font-size : 10pt }

-->

</style>

CSS 別ファイル



名前

宮口田 射太郎

学部

情報工学部

css File

```
#myNameFont{
  color:blue;
  font-family:fantasy
}

.myHeading{
  font-size:18pt;
  color:white;
  background-color:black
}
```

html File

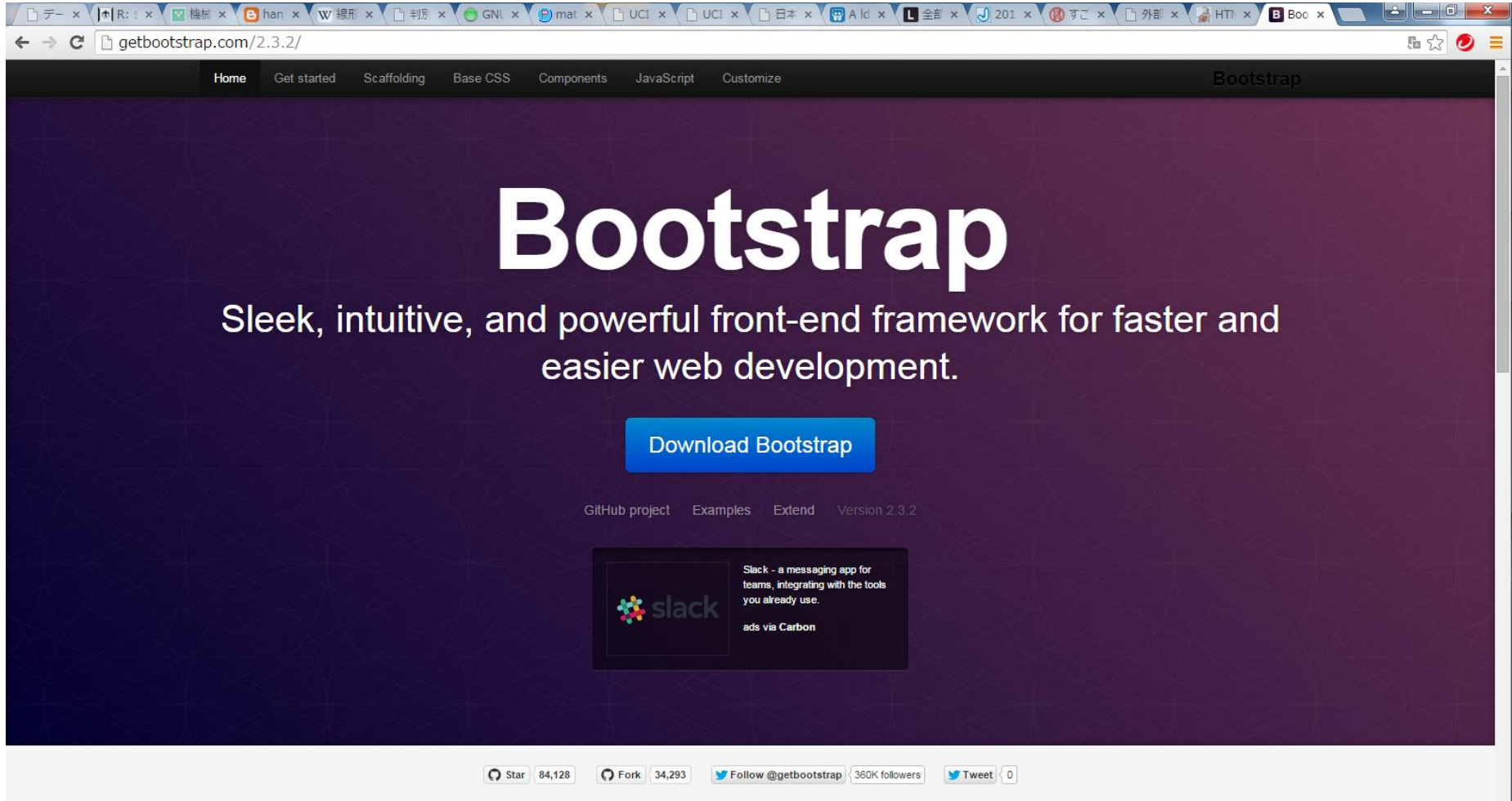
```
<html>
<head>
  <LINK href="sample_style.css" rel="stylesheet"
  type="text/css">
</head>

<body>
  <h2 class = "myHeading"> 名前 </h2>
  <p id="myNameFont"> 宮口田 射太郎</p>
  <h2 class = "myHeading"> 学部 </h2>
  <p> 情報工学部 </p>
</body>
</html>
```

BootStrap

いい感じのデザインのCSSファイルの集まり

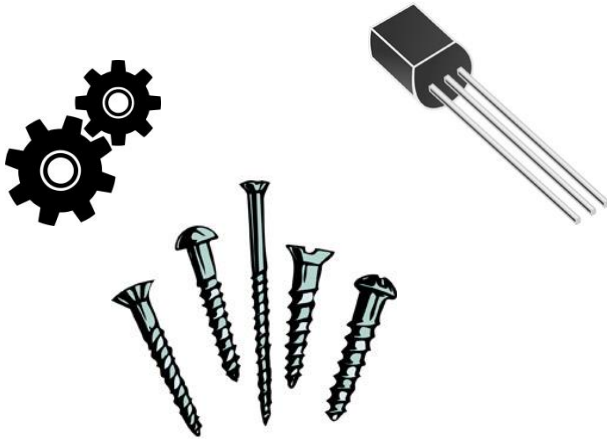
Bootstrapにアクセス



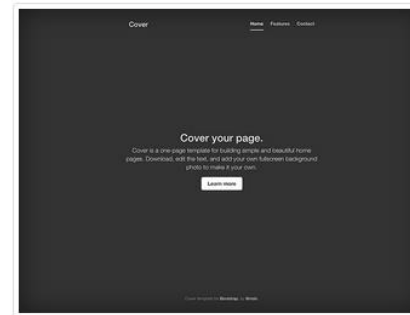
BootStrap

<いい感じのページ>

構成するパーツを提供

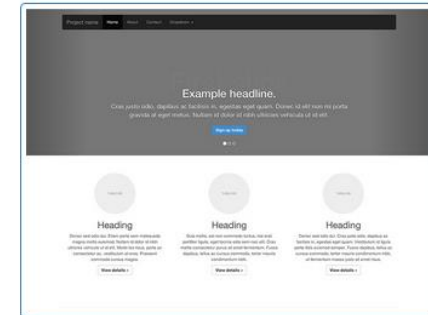


Custom components



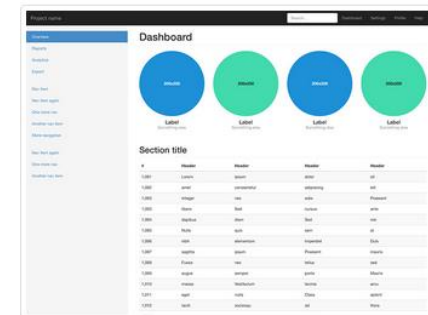
Cover

A one-page template for building simple and beautiful home pages.



Carousel

Customize the navbar and carousel, then add some new components.



しかも

組み立てなくてもいい感じのサンプルがある！

```
<html lang="en">
<head>
  <link href="css/bootstrap.min.css" rel="stylesheet">
</head>

<body>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <script src="js/bootstrap.min.js"></script>

  <div id="myCarousel" class="carousel slide">
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
    </ol>
    <!-- Carousel items -->
    <div class="carousel-inner">
      <div align="center" class="active item"></div>
      <div align="center" class="item"></div>
      <div align="center" class="item"></div>
    </div>
    <!-- Carousel nav -->
    <a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>
    <a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a>
  </div>
</body>
</html>
```

赤字 が用意されている部品

見てみよう

Java Script

JavaScript

<script> ~ </script>の中にプログラムを書ける。

- Userの行動に対して、
アクションのあるウェブページの作成が可能に

具体的に？

fncCall



fncCall



fncCall2



```
<html>
<head>
</head>
```

```
<body>
<script>
var gl_count = 0; //グローバル変数
function fncCall(){
  var mes = "関数が呼び出されました！"; //ローカル変数
  gl_count = gl_count + 1;
  alert( gl_count + "回" + mes );
}
</script>
```

```
<script>
function fncCall2(){
  gl_count = 0;
  fncCall();
}
</script>
```

```
<button type="button" onclick="fncCall()">fncCall</button>
<button type="button" onclick="fncCall2()">fncCall2</button>
```

```
</body>
</html>
```

```
<html>
<head>
</head>

<body>
<script>
var gl_count = 0; //グローバル変数 //変数の指定はvar を使う

function fncCall(){ // 関数1
  var mes = "関数が呼び出されました！"; //ローカル変数
  gl_count = gl_count + 1;
  alert( gl_count + "回" + mes );
}
</script>

<script>
function fncCall2(){ // 関数2
  gl_count = 0; // グローバル変数なので他にも影響
  fncCall(); //別のスクリプトからも関数呼び出し可能
}
</script>

<button type="button" onclick="fncCall()">fncCall</button>
<button type="button" onclick="fncCall2()">fncCall2</button>

</body>
</html>
```

`<script> ~ </script>`の中であれば
C言語のように関数を実行することが
できる。

```
<script>
  fncCall()
</script>
```

← こんな
感じで

return 変数名;
を指定することで返り値
※関数に返り値の型指定はない

※注意: グローバル変数は無名
関数の中では変更されない。

無名関数

```
Func1(Func2()){  
    .....  
}
```

```
Func2(){  
    .....  
    return ret;  
}
```

```
Func1(function (){  
    .....  
    return ret;  
}){  
    .....  
}
```

jQuery

いい感じのJS関数ファイルの集まり

JavaScript

```
<html>
<head>
</head>

<body>
  <h1>PopUp Image</h1>
  <div class="waku">
    
    
    
  </div>

  <script language="JavaScript"><!--
    function up(elm){
      elm.style.marginTop = "10px";
      elm.style.marginBottom = "30px";
    }
    function down(elm){
      elm.style.marginTop = "20px";
      elm.style.marginBottom = "20px";
    }
  //-->
</script>
</body>
</html>
```

自作関数

JQuery

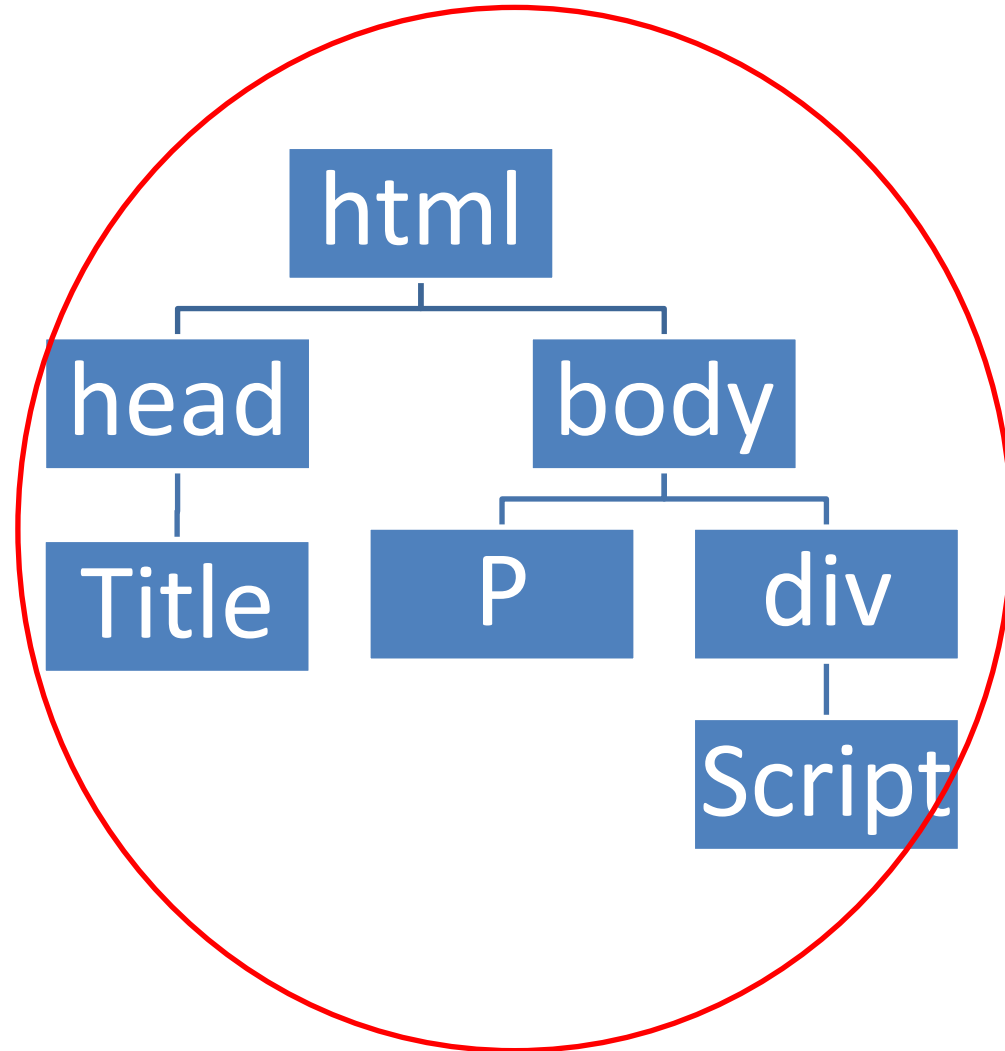
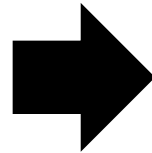
```
<html>
<head>
  <script type="text/javascript" src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.7.1.min.js"></script>
  <script language="JavaScript"><!--
    $(function(){
      $(".pop").hover(
        function(){$(this).css("margin-top","10px").css("margin-bottom","30px")},
        function(){$(this).css("margin-top","20px").css("margin-bottom","20px")}
      );
    });
  //--></script>
</head>

<body>
  <h1>PopUp Image</h1>
  <div class="waku">
    
    
    
  </div>
</body>
</html>
```

jQueryで与えられる関数

DOM

```
<HTML>  
  
  <HEAD>  
    <TITLE>~</TITLE>  
  </HEAD>  
  
  <BODY>  
    <p></p>  
    <div>  
      <script></script>  
    </div>  
  
  </BODY>  
</HTML>
```



※JavaScriptではこっちをいじる