

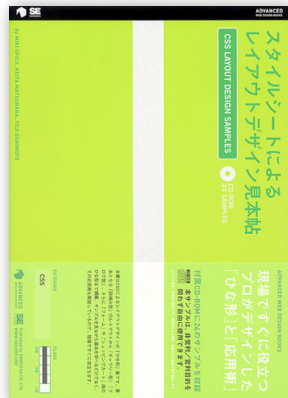
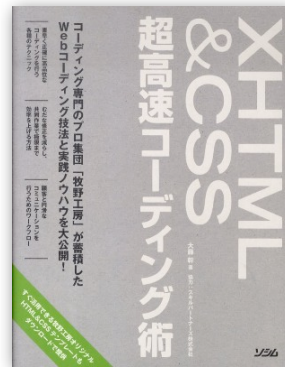


LPI-Japan主催

HTML5 プロフェッショナル認定試験 レベル1 ポイント解説 無料セミナー

講師：大藤 幹

(株式会社クリーク・アンド・リバー社 認定講師)



- HTML5プロフェッショナル認定試験 試験概要
- これだけは覚えておくべきHTML5の仕様
- これだけは覚えておくべきCSS3の仕様
- これだけは覚えておくべきその他の基礎知識

HTML5プロフェッショナル認定試験

試験概要



2つのレベル



Level.1

HTML 5 Level.1

→ Markup Professional

マルチデバイスに対応した静的なWebコンテンツを HTML5を使ってデザイン・作成できる。



Level.2

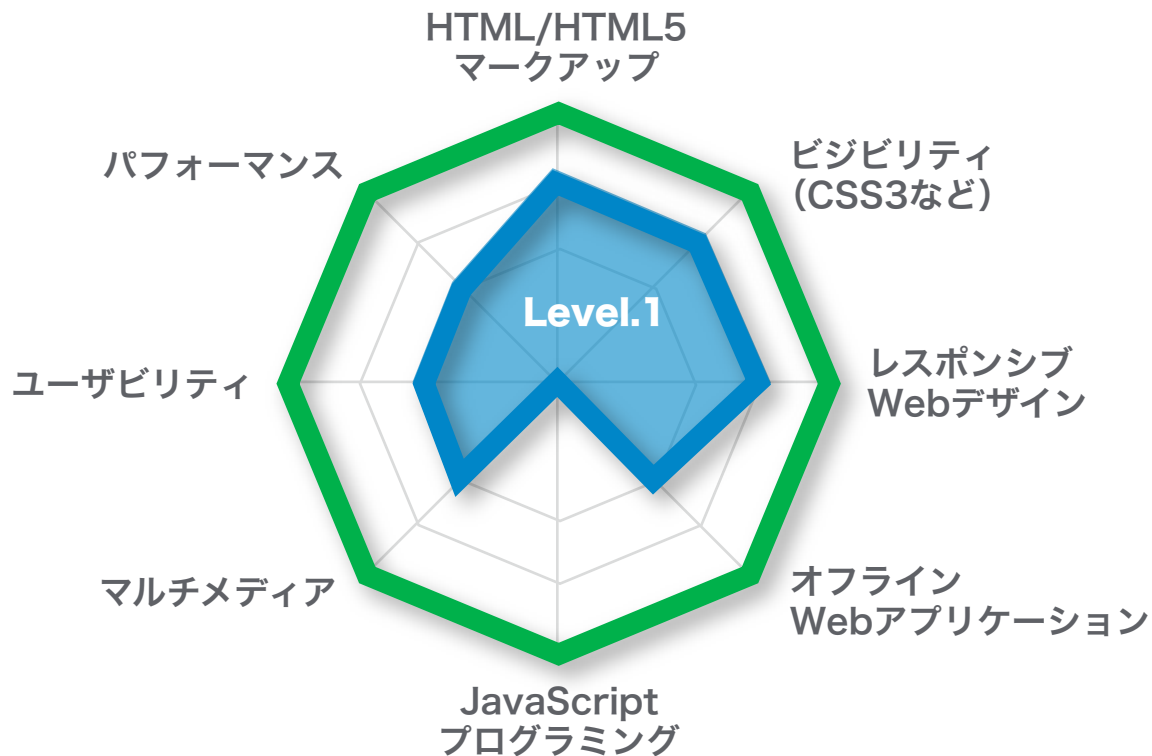
HTML 5 Level.2

→ Application Development Professional

システム間連携や最新のマルチメディア術に対応したWebアプリケーションや動的Webコンテンツの開発・設計ができる。

Level.1 

Level.2 





レベル1 試験概要



所要時間	90分 (機密保持契約とアンケートの時間を含む)
試験問題数	約60問
有意性の期限	5年間
受験料	¥15,000 (税抜)

Webの基礎知識

30%

CSS3

20%

要素

37%

レスポンシブWebデザイン

10%

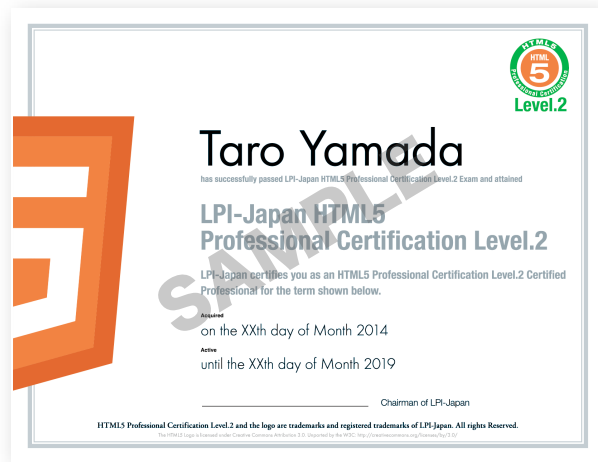
オフラインWebアプリケーション (概要とマニフェスト)

3%



認定証・認定カード

認定証



認定カード



認定者ロゴ (名刺用)





オフィシャルサイト

<http://www.html5exam.jp/>

Twitter

@html5Cert

Facebook

<https://www.facebook.com/html5exam>

受験予約

<http://www.pearsonvue.com/japan/Registration/>

これだけは覚えておくべき

HTML5の仕様

- **template**要素 → 内容をスクリプトで挿入する範囲
- **rb**要素 → **ruby base text**
- **rtc**要素 → **ruby text container**

※HTML5の要素は合計**111種類**となった

```
<table>
<tr><th>製品名</th><th>サイズ</th><th>色</th></tr>
<template id="row">
<tr><td></td><td></td><td></td></tr>
</template>
</table>
```

※<http://www.w3.org/TR/html5/scripting-1.html#the-template-element>

HTML5文書の 全体構造

```
<!DOCTYPE html>  
<html lang="ja">  
<head>  
<meta charset="UTF-8">  
<title>○○○</title>  
  
</head>  
<body>  
  
</body>  
</html>
```

`<link rel="stylesheet" href="style.css">`

`<style>`

• • •

`</style>`

`type="text/css"`



`<script>`

• • •

`</script>`

`type="text/javascript"`



HTML5で 削除された要素

HTML5で削除された要素1

要素名	要素のあらわすもの・役割
acronym	頭字語（各単語の頭文字を続けてできた略語）
applet	Javaアプレット
basefont	基準フォントの種類・色・サイズの指定
big	大きな文字
center	中央揃え
dir	ディレクトリ・リスト
font	フォントの種類・色・サイズの指定

要素名	要素のあらわすもの・役割
frame	フレームの内容を指定
frameset	フレームの区切り方の定義
isindex	1行の入力フィールド
noframes	フレームが表示できない環境用の内容
strike	取消線を引いて表示
tt	等幅フォントで表示

HTML5で 変更された要素

HTML5で役割や意味が変更された要素1

要素名	要素のあらわすもの・役割
b	注目してほしい部分を示す。強調や重要などの意味合いを示すものではない。キーワードや製品名部分などに使用する。
cite	その部分が作品のタイトルであることを示す（HTML4では出典や参照先を意味していた）。人名には使用しない。
dl	用語解説型リスト。HTML4ではdefinition Listの略だったが、HTML5ではdescription Listの略。定義の意味はなくなった。
hr	段落単位での内容の区切りを示す。セクションの中で、話題や場面が変わる部分などで使用することを想定。
i	一般に慣習としてイタリックで表記される部分。生物の学名や船の名前、言語の種類が異なっている部分などに使用する。

要素名	要素のあらわすもの・役割
s	その部分を取り消したことを示す。すでに正しくない情報となってしまった部分や、無関係となった部分などに使用。
small	メインコンテンツには含まれない部分において、一般に小さな文字で掲載される部分。Copyrightの情報などに使用。
strong	その部分が重要であることを示す。入れ子にすることで、重要さの度合いを高めることができる。
u	中国語の固有名詞とそうでない部分を区別したり、スペルが間違っている部分を示したい場合などに使用。

HTML5で 追加された要素

要素名	要素のあらわすもの・役割
section	章や節のようなセクションを示す（他のセクション要素が適当でないセクションにはこれを使用する）。
article	ニュースの記事やブログの記事のように、内容がそれだけで完結しているセクションを示す。ブログの各コメントにも使用する。
aside	メインコンテンツには含まれていない部分のセクションを示す。広告やリンクのリスト、補足記事などのセクションに使用する。
nav	グローバル・ナビゲーションなど、主要なナビゲーションであることを示す。主要でないナビゲーションには使用しない。

セクションのタグがなくても、見出しのタグがあれば、そこにセクションがあるものと認識する。

見出しの階層が前の見出しと同じか高ければ、前のセクションを終了して新しいセクションを開始する。

見出しの階層が前の見出しよりも低ければ、今のセクションの中でサブセクションを開始する。

要素名	要素のあらわすもの・役割
header	Webページのいわゆるヘッダー部分を示す要素。一般に、ロゴ画像や検索フォーム、ナビゲーションなどを含む部分。
main	Webページのメインコンテンツ部分を示す要素。各ページで共通するロゴ画像・検索フォーム・ナビゲーション・著作権情報などは含めない。
footer	そのfooter要素のあるセクションに関する情報を入れる要素。下にあっても上にあってもOK。

要素名	要素のあらわすもの・役割
bdi	双方向テキストの中での分離・独立 (isolate) された範囲。
mark	テキストの特定の部分を目立つようにして (ハイライト表示にして) 注目してもらいたい場合に使用。
time	コンピュータが読み取り可能な形式の日付・時刻を <code>datetime</code> 属性の値または要素内容で示す。
data	要素内容をコンピュータが読み取り可能な形式にしたデータを <code>value</code> 属性 (必須) の値で示す。
wbr	英単語の途中など、通常は行の折り返しが行われない範囲において、改行可能な位置を示す。空要素。

要素名	要素のあらわすもの・役割
ruby	ルビをふる範囲全体
rb	ruby base text → ルビをふる漢字部分 ※新要素
rt	ruby text → ルビ・ふりがな
rtc	ruby text container → rt要素のグループ ※新要素
rp	ruby parentheses → ルビ未対応用のカッコ

め ま ん べ つ ← **rt** (ruby text) 要素

女 満 別 ← **rb** (ruby base text) 要素

<ruby>

<rb>女満別</rb>

<rt>めまんべつ</rt>

</ruby>

`<ruby>`

女`<rb>`満`</rb>``<rb>`別`</rb>`

`<rt>`め`</rt>``<rt>`まん`</rt>``<rt>`べつ`</rt>`

`</ruby>`

め まん べつ
女 満 別
me man betsu



<ruby>

女<rb>満</rb><rb>別</rb>

<rt>め</rt><rt>まん</rt><rt>べつ</rt>

<rtc>

<rt>me</rt><rt>man</rt><rt>betsu</rt>

</rtc>

</ruby>

<ruby>

<rb>女満別</rb>

<rp> (</rp><rt>めまんべつ</rt><rp>) </rp>

</ruby>



女満別 (めまんべつ)

要素名	要素のあらわすもの・役割
figure	メインコンテンツから参照される図表・図版（写真やソースコードなども含む）
figcaption	figure要素のキャプション

要素名	要素のあらわすもの・役割
audio	音声データを組み込む要素。音声データのURLをsrc属性に指定。
video	動画データを組み込む要素。動画データのURLをsrc属性に指定。
source	形式の異なる複数のデータを指定したい場合に使用。
track	同期させるテキスト・トラック（字幕など）を指定する要素。
embed	プラグインを使用するデータを組み込むための要素。
canvas	ビットマップの動的グラフィックを表示。JavaScriptで制御。

要素名	要素のあらわすもの・役割
meter	ある範囲のどの位置かを示すフォーム部品（メーター）
progress	タスクの進行状況を示すフォーム部品（プログレスバー）
datalist	input要素の選択肢（サジェスト機能のように入力候補を表示）
output	他のフォーム部品からの計算結果を表示させるフォーム部品 ※ at risk
keygen	公開鍵暗号における公開鍵・秘密鍵のペアを生成するフォーム部品

要素名	要素のあらわすもの・役割
details	特定の操作（▶をクリックなど）で表示される詳細情報 ※ at risk
summary	details要素で最初から表示させる見出しや説明などの情報 ※ at risk
dialog	インタラクティブな操作のためのダイアログボックスやウィンドウ、インスペクタなど ※ at risk
template	内容をスクリプトで挿入する範囲 ※ 新要素

属性関連 の追加と変更

属性名	値
id	固有の名前
class	種類を示す名前
title	補足情報
lang	言語コード
style	CSSの「プロパティ: 値;」
accesskey	ショートカットキー
tabindex	タブキーによる移動の順序
dir	文字表記の方向 (ltr/rtl/auto)

属性名	値
contenteditable	編集の可・不可 (true/false)
draggable	ドラッグの可・不可 (true/false/auto)
dropzone	ドロップ時の挙動 (copy/move/link)
spellcheck	スペルチェックをするかどうか (true/false)
translate	ローカライズの際に翻訳するかどうか (yes/no)
hidden	非表示 (指定するとtrue/指定しないとfalse)

- データを格納するにふさわしい要素や属性がない場合に作成できる独自の属性
- サイトの内部のスクリプトなどで利用することを想定して用意されたもの
 - ※ サイトの外部から利用するためのものではない

- 属性名は必ず「**data-**」ではじめる
- 「**data-**」のあとに1文字以上必要
- どの要素にいくつでも、任意の値で指定可能

※ <http://www.w3.org/TR/html5/dom.html#global-attributes>

<li data-length="11m32s">弦楽四重奏曲第1番

<li data-length="08m55s">弦楽四重奏曲第2番

<li data-length="22m04s">弦楽四重奏曲第3番

<li data-length="13m46s">弦楽四重奏曲第4番

<li data-length="30m00s">弦楽四重奏曲第5番

※各曲の長さをdata-length属性であらわした例

※曲の長さで並べ替えることなどが可能となる

これだけは覚えておくべき

CSS3の仕様

セレクタ

スタイルの適用先を示す部分がセレクトア



```
div#wrapper {  
  width: 900px;  
  padding: 1em;  
  . . .  
}
```



```
セレクトア {  
  プロパティ: 値;  
  プロパティ: 値;  
  . . .  
}
```

CSS2.1

[属性名]

[属性名 = "属性値全体"]

[属性名 ~ = "属性値の一つ"]

[属性名 | = "属性値の-の前"]

CSS3

[属性名 ^ = "開始文字列"]

[属性名 \$ = "終了文字列"]

[属性名 * = "含まれる文字列"]

CSS2.1

- `:first-letter`
- `:first-line`
- `:before`
- `:after`

CSS3

- `::first-letter`
- `::first-line`
- `::before`
- `::after`

CSS2.1

セレクタA セレクタB

※Aの中に含まれるB

セレクタA > セレクタB

※Aの子要素であるB

セレクタA + セレクタB

※Aの直後にあるB

CSS3

セレクタA ~ セレクタB

※Aよりも後にあるB

:first-child

最初の要素に適用 **CSS2.1**

:first-of-type

最初の要素に適用 (同一要素)

:last-child

最後の要素に適用

:last-of-type

最後の要素に適用 (同一要素)

※「nth」は「○th」、つまり「4th」や「5th」のような「n番目」の意味

:nth-child(式)

前からn番目に適用

:nth-last-child(式)

後ろからn番目に適用

:nth-of-type(式)

前からn番目に適用 (同一要素)

:nth-last-of-type(式)

後ろからn番目に適用 (同一要素)

:nth-child(7)

7番目の要素に適用

:nth-child(2n+1)

奇数番目の要素に適用 (an+b形式)

:nth-child(2n+0)

偶数番目の要素に適用 (an+b形式)

:nth-child(odd)

奇数番目の要素に適用 (キーワード)

:nth-child(even)

偶数番目の要素に適用 (キーワード)

:only-child

一人っ子の要素に適用

:only-of-type

一人っ子の要素に適用 (同一要素)

:disabled

disabled状態の要素に適用

:enabled

disabled状態ではない要素に適用

:checked

checked状態の要素に適用

:target

「#○○○」リンクのジャンプ先に適用

:root

ルート要素 (html要素) に適用

:empty

内容が空の要素に適用

:not(セレクト)

「セレクト」に該当しない要素に適用

セレクトタによる 優先順位

- CSSの表示指定が競合した場合の優先度はセレクトア
で決められる
- セレクトアを使用しない「style属性」による指定は最
優先される

セレクタによる優先順位の計算方法

セレクタに含まれる次のものをそれぞれカウントして3桁の数字を作成
その3桁の数字が大きなものほど優先順位が高くなる

IDセレクタの数



属性関連セレクタの数

(クラスセレクタ、属性セレクタ、
疑似クラスの数)



要素関連セレクタの数

(タイプセレクタ、疑似要素の数)



※ユニバーサルセレクタ (*) はカウントしない

:not(セレクト)

- :not() 自体は疑似クラスとしてはカウントしない
- () 内のセレクトだけを普通のセレクトとしてカウント

セレクタによる優先順位の計算の例1

div



ID	属性	要素
0	0	1

.inner



ID	属性	要素
0	1	0

div.inner



ID	属性	要素
0	1	1

#logo



ID	属性	要素
1	0	0

div#logo



ID	属性	要素
1	0	1

body div#logo



ID	属性	要素
1	0	2

色と半透明の指定方法

rgb(Red, Green, Blue)



rgba(Red, Green, Blue, 不透明度 Alpha)



```
color: rgba( 255, 255, 255, 0.2 );
```

```
color: rgba( 255, 255, 255, 0.4 );
```

```
color: rgba( 255, 255, 255, 0.6 );
```

```
color: rgba( 255, 255, 255, 0.8 );
```

```
color: rgba( 255, 255, 255, 1.0 );
```


hsla(Hue, Saturation, Lightness, Alpha)



色相

0



360



彩度

0%



100%



明度

0%



100%

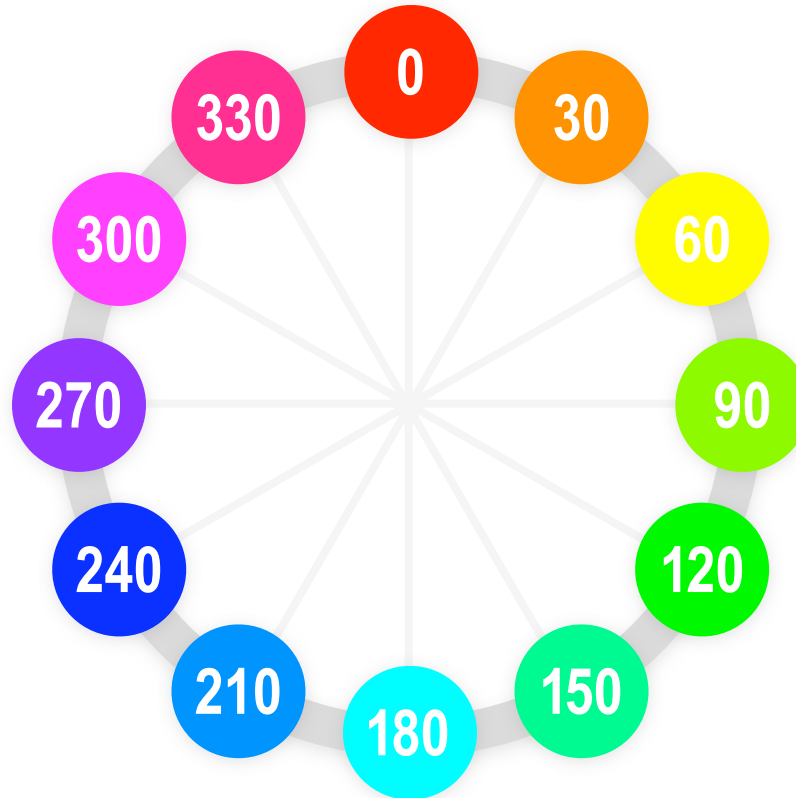


不透明度

0.0



1.0



```
#box01{ opacity: 0.1 };
```

```
#box02{ opacity: 0.2 };
```

```
#box03{ opacity: 0.3 };
```

```
#box04{ opacity: 0.4 };
```

```
#box05{ opacity: 0.5 };
```

```
#box06{ opacity: 0.6 };
```

```
#box07{ opacity: 0.7 };
```

```
#box08{ opacity: 0.8 };
```

```
#box09{ opacity: 0.9 };
```

```
#box10{ opacity: 1.0 };
```

回轉・移動・拡大縮小

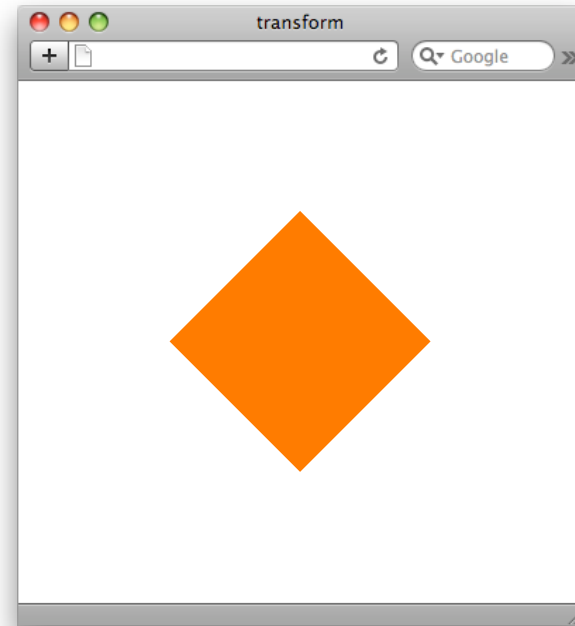
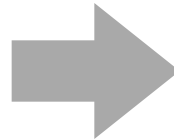
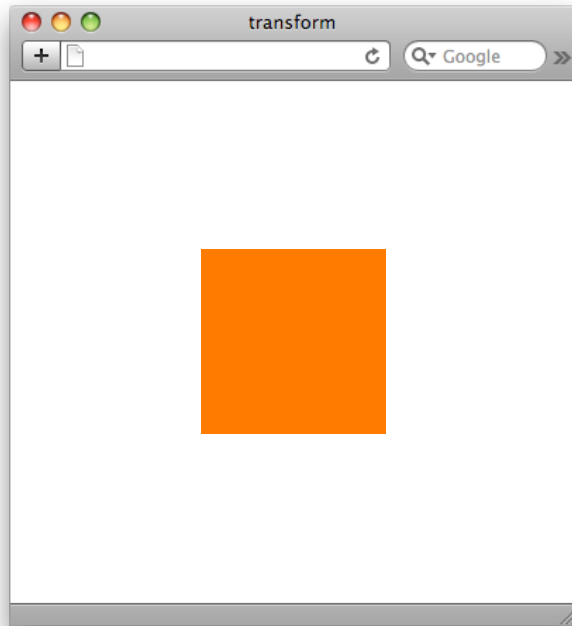
transform: rotate(45deg);

transform: translate(50px);

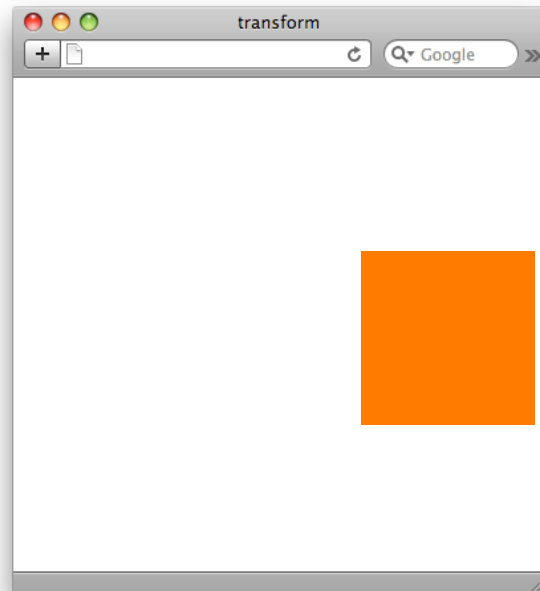
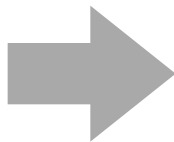
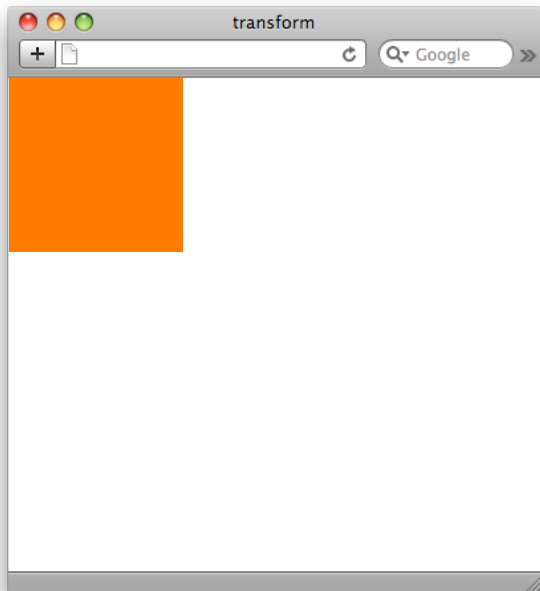
transform: scale(1.5);



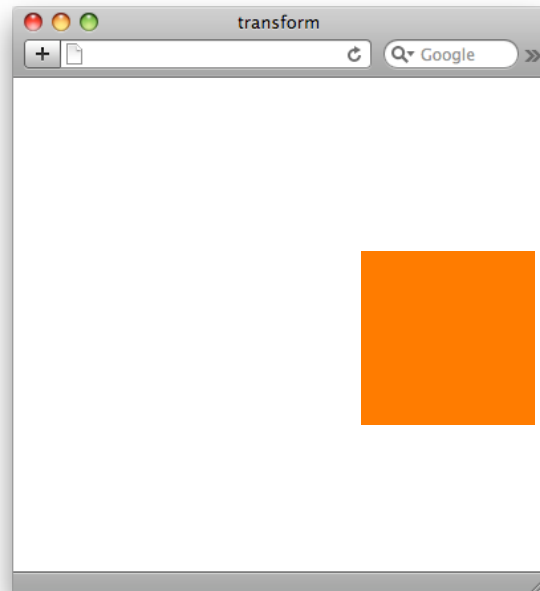
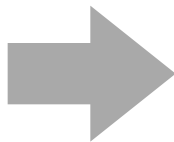
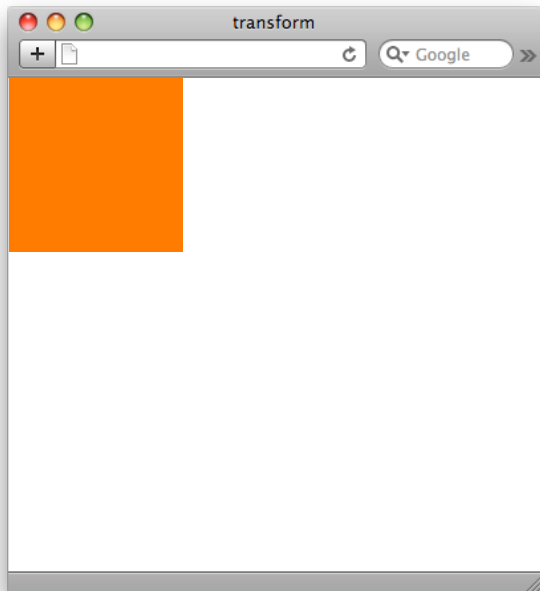
transform: rotate(45deg);



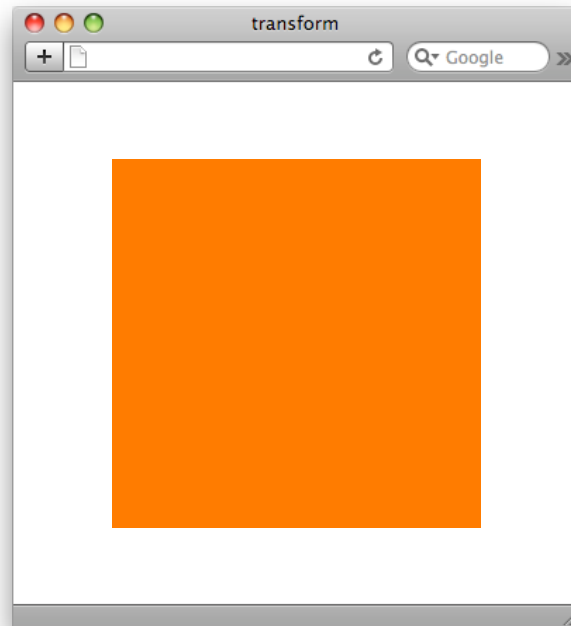
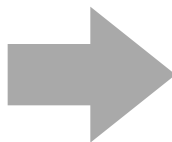
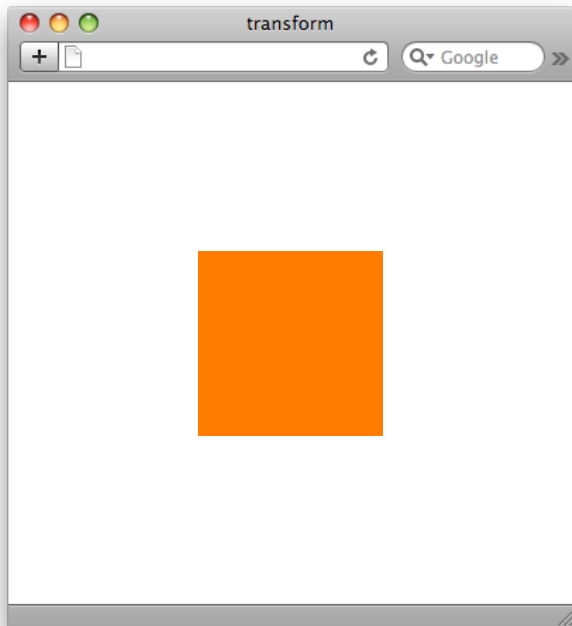
transform: translate(100px, 50px);



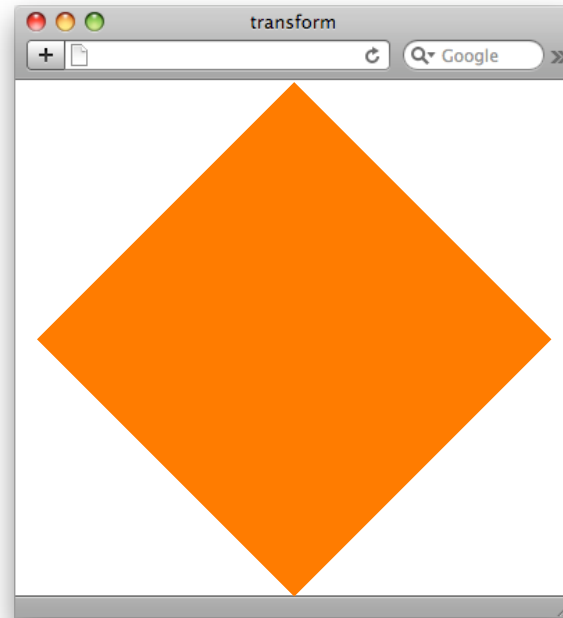
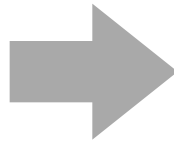
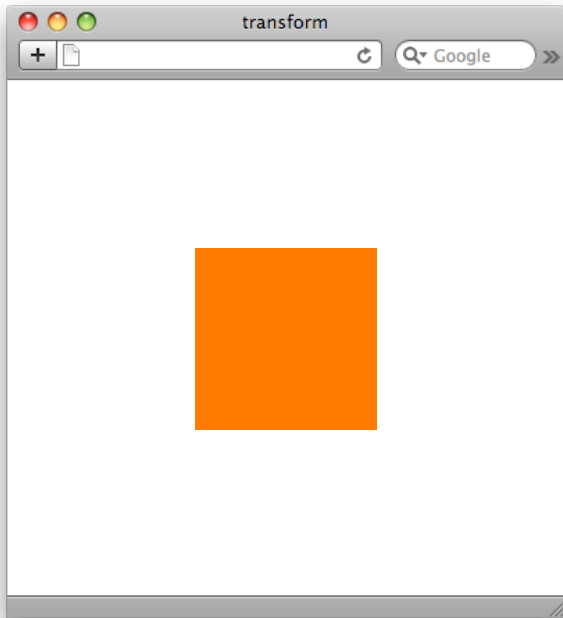
transform: translate(100px, 50px);
= position: relative; left: 100px; top: 50px;



transform: **scale(2.0);**



transform: scale(2) rotate(45deg);



トランジション

Demo

アニメーション

Demo

これだけは覚えておくべき

その他の基礎知識

メディアクエリ

HTML

```
<link media="screen" . . . >
```

```
<style media="screen" . . . >
```

```
. . .
```

```
</style>
```

CSS

```
@media screen {
```

```
. . .
```

```
}
```

HTML

```
<link media="screen かつ、幅が600px以上" . . . >
```

```
<style media="screen かつ、幅が1024px以下" . . . >
```

```
. . .
```

```
</style>
```

CSS

```
@media screen かつ、画面が縦長の状態 {
```

```
. . .
```

```
}
```

メディア **and** (メディア特性: 値)

メディア **and** (メディア特性: 値) **and** (メディア特性: 値) …

not メディア and (メディア特性: 値) ...

※条件を逆にする

only メディア and (メディア特性: 値) ...

※古いブラウザ対策

メディア特性とその値 (抜粋)

メディア特性	説明	値
width	表示領域の幅 (スクロールバーも含む)	実数 + 単位
min-width	表示領域の最小の幅 (スクロールバーも含む)	実数 + 単位
max-width	表示領域の最大の幅 (スクロールバーも含む)	実数 + 単位
height	表示領域の高さ (スクロールバーも含む)	実数 + 単位
device-width	出力メディアの画面の幅	実数 + 単位
min-device-width	出力メディアの画面の最小の幅	実数 + 単位
max-device-width	出力メディアの画面の最大の幅	実数 + 単位
resolution	出力メディアの解像度	実数 + 単位
orientation	縦長・縦横同じ (portrait) / 横長 (landscape)	portrait, landscape

dpi → **dots per inch**

dpcm → **dots per centimeter**

```
<link media="screen and (max-width: 1000px)" ... >
```

※表示領域の幅が1000px以下の時に適用

```
<link media="only screen and (min-width: 500px)  
and (max-width: 1000px)" ... >
```

※表示領域の幅が500px以上、1000px以下の時に適用

オフラインWebアプリケーション

キャッシュマニフェスト

Webアプリケーション (Webサイト) を オフラインでも動作させる仕組み

※<http://www.w3.org/TR/html5/browsers.html#offline>

キャッシュするファイル、キャッシュしないファイル、
その他関連情報の指示ファイル (テキスト形式)

※<http://www.w3.org/TR/html5/browsers.html#manifests>

CACHE MANIFEST ←

必ず1行目に記述

CACHE: ←

styles/default.css
images/background.png

キャッシュさせて使うファイルを指定

NETWORK: ←

comm.cgi

常にネットから取得するファイルを指定

FALLBACK: ←

offline.html

取得失敗時の代替ファイルを指定

SETTINGS: ←

fast ←

キャッシュ (モード) の設定

キャッシュを使う (デフォルト)

CACHE MANIFEST

#この行はコメントです ← 「#」で始まる行はコメント

styles/default.css ←
images/background.png ← 省略すると「CACHE:」

NETWORK:

* ← 「CACHE:」で指定されていない
ファイル全部

SETTINGS:

prefer-online ← 可能ならネットからファイルを取得

FALLBACK:
offline.html

```
<!DOCTYPE html>
```

```
<html lang="ja" manifest="○○○.appcache">
```

```
  ~
```

```
</html>
```

- 拡張子 → **.appcache**
- MIMEタイプ → **text/cache-manifest**
- エンコーディング → **UTF-8**

LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.