

## HTML/CSS実習

# 09. ボックスのスタイル

株式会社ジードライブ

# 今回学ぶこと

---

- `display`プロパティ
- ボックスモデル
- ボックスに対するスタイル
  - 幅、高さ、パディング、境界線、マージン
  - 背景色や背景画像の設定

# displayプロパティ

- 要素の表示形式に関するプロパティで、以下のような値を指定することができる

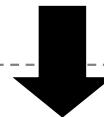
値	説明
block	<ul style="list-style-type: none"><li>縦に並ぶ（前後に改行を伴う）</li><li>幅や高さの指定が可能</li></ul>
inline	<ul style="list-style-type: none"><li>横に並ぶ（前後に改行を伴わない）</li><li>幅や高さの指定ができない</li></ul>
inline-block	<ul style="list-style-type: none"><li>横に並ぶ（前後に改行を伴わない）</li><li>幅や高さの指定が可能</li></ul>
none	<ul style="list-style-type: none"><li>表示されない</li><li>スマートフォンとPCで表示を切り分けたい場合などに利用する</li></ul>

# display: block;

- 見出し、段落、ヘッダー、フッター、div等の初期値
- 前後に改行を伴う⇒縦に並ぶ

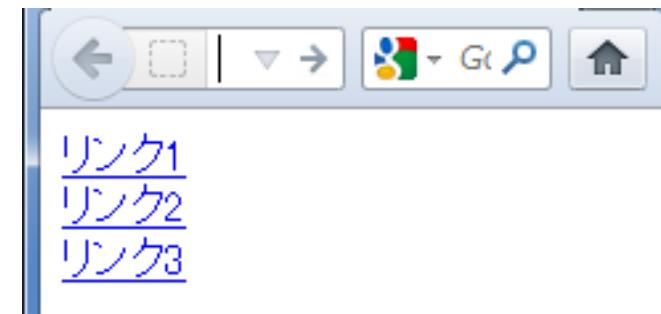
a要素はdisplay:inline;が初期値なので、表示すると横に並ぶ

```
<a href="#">リンク1</a>
<a href="#">リンク2</a>
<a href="#">リンク3</a>
```



a要素にdisplay:block;を設定すると、前後に改行を伴い縦に並ぶ

```
a {
  display: block;
}
```

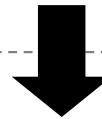
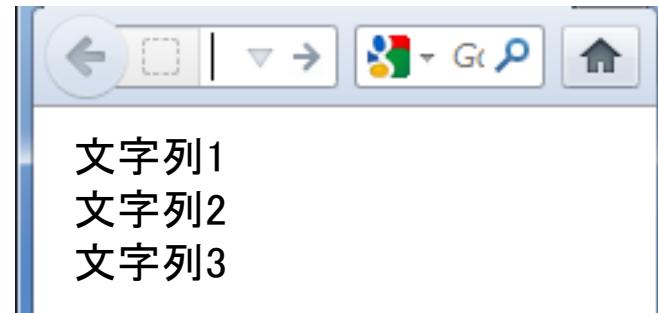


# display: inline;

- a要素、span要素などの初期値
- 前後に改行を伴わない⇒横に並ぶ

p要素はdisplay:block;が初期値なので、表示すると縦に並ぶ

```
<p>文字列1</p>
<p>文字列2</p>
<p>文字列3</p>
```



p要素にdisplay:inline;を設定すると、横に並ぶ

```
p {
  display: inline;
}
```



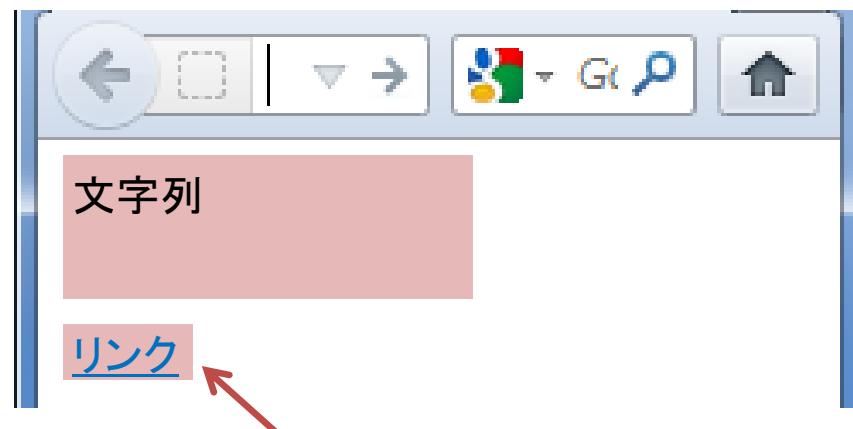
# 幅／高さ設定の可否

- `display:block;` の場合、幅や高さの設定が可能
- `display:inline;` の場合、幅や高さの設定が不可能

`p`要素には幅や高さの設定が適用されるが、`a`要素には適用されない

```
<p>文字列</p>
<a href="#">リンク1</a>
```

```
p, a {
  width: 100px;
  height: 30px;
  background: pink;
}
```



幅や高さの設定がきいていない

# display: inline-block;

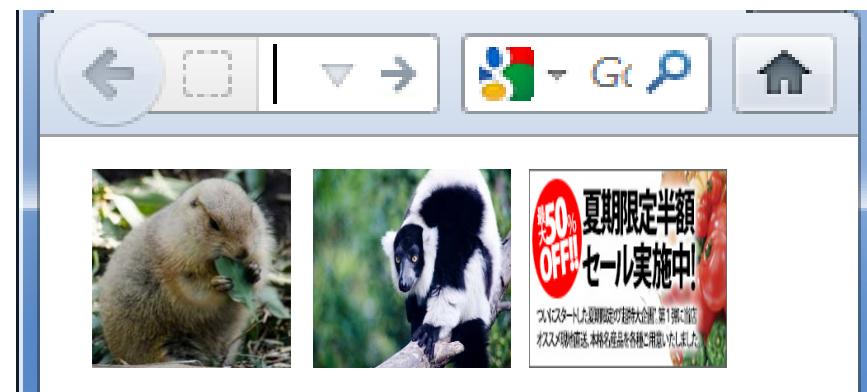
- img要素、input要素などの初期値
- inlineのように横に並び、blockのように幅や高さを設定することができる

img要素には幅や高さの設定が適用されるが、横に並ぶ

```
  
  

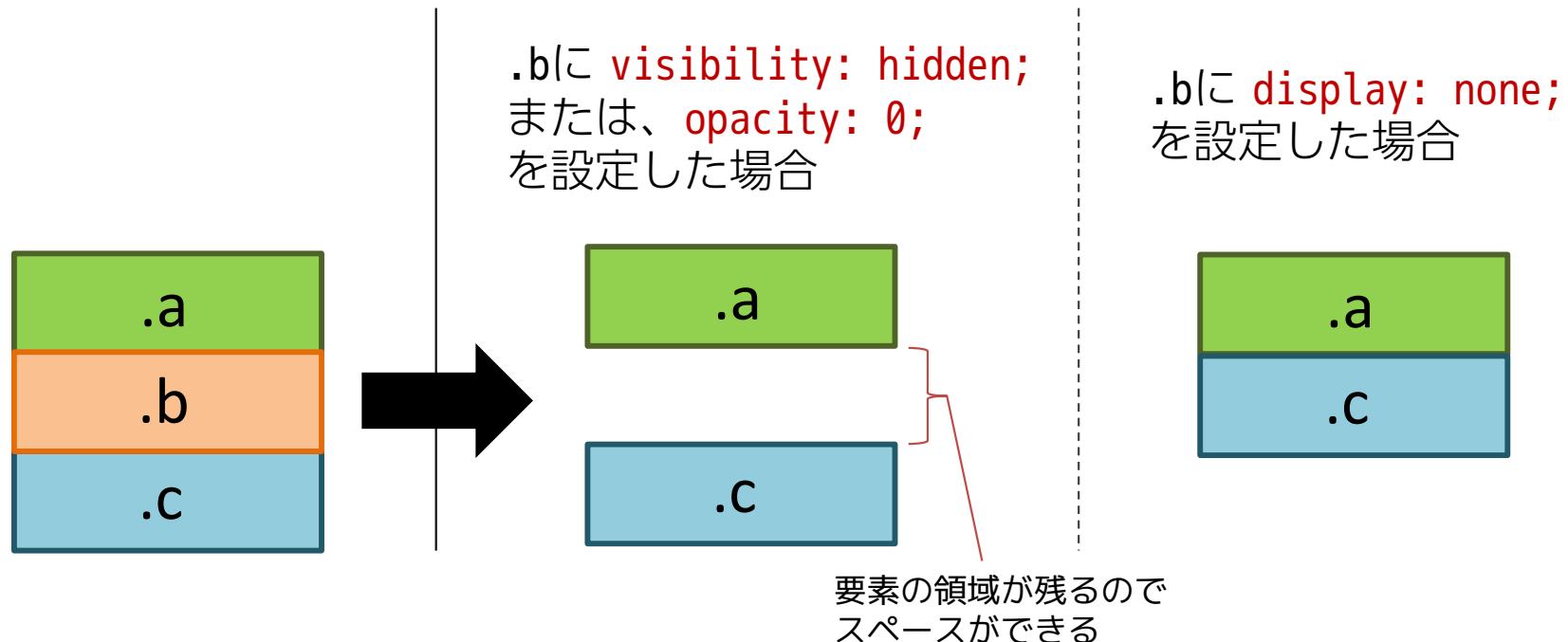
```

```
img {  
    width: 30px;  
    height: 30px;  
}
```



# display: none;

- 要素が非表示になり、領域もなくなる
  - HTML自体がなくなるわけではないので、開発者ツールを使えば見ることができる
  - visibility:hidden; や opacity:0; では透明になるだけで、要素の領域は残る



# 練習

---

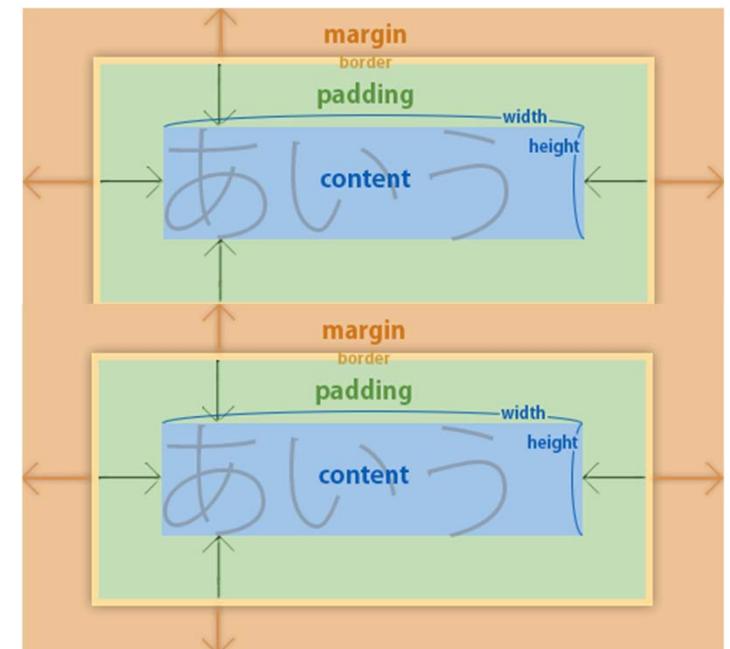
- 練習09-1

# ボックスモデル

# ボックスモデルとは

- HTMLの各要素を矩形の領域（ボックス）で捉える考え方
  - ボックスは、内容領域・パディング・ボーダーから成る
  - ボックス同士の間隔をマージンという

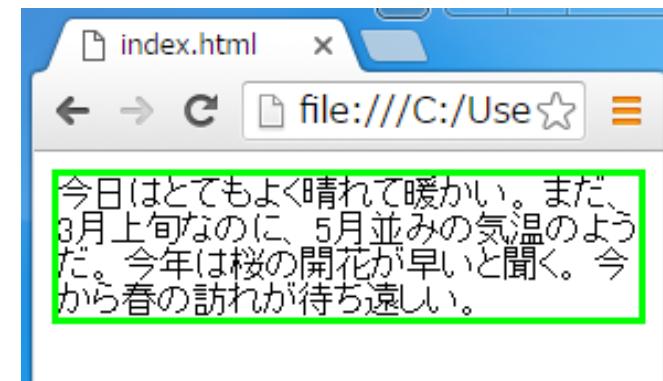
用語	説明
内容の領域 Content Area	要素を表示するための領域 width, heightプロパティで任意の幅と高さを指定可能
padding	内容領域とborderの間の余白 背景色や背景画像が表示される
border	ボックスの境界線 太さ、線種、色を指定することができる
margin	他のボックスとの間隔



# 境界線 border

- 境界線の幅、線のスタイル、色を半角アキで指定する
  - 線幅、線種、色の記述は順不同。線種の指定は必須

```
<p>  
今日はとてもよく晴れて暖かい。  
まだ、3月上旬なのに、  
5月並みの気温のようだ。  
今年は桜の開花が早いと聞く。  
今から春の訪れが待ち遠しい。  
</p>
```



```
p {  
    border: 3px solid #0F0;  
}
```

線幅 線のスタイル 色

# 線のスタイル

- 線のスタイルには以下のようなものがある

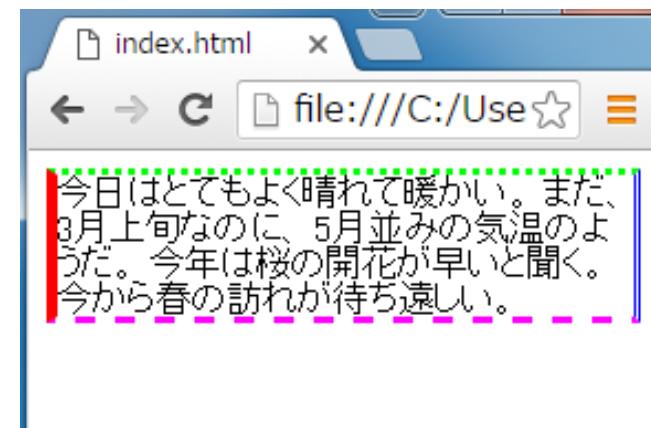
値	説明	表示
none	非表示(初期値)	
hidden	非表示 (重複時優先)	
solid	実線	—
double	2本線	==
dotted	点線	.....

値	説明	表示
dashed	破線	-----
outset	隆起した立体影	
inset	くぼんだ立体影	
ridge	隆起した線	
groove	くぼんだ線	

# 個別に設定する（上下左右）

- サブプロパティとして、**-top, -bottom, -left, -right**を付けることで、上下左右のボーダーを個別に設定することができる

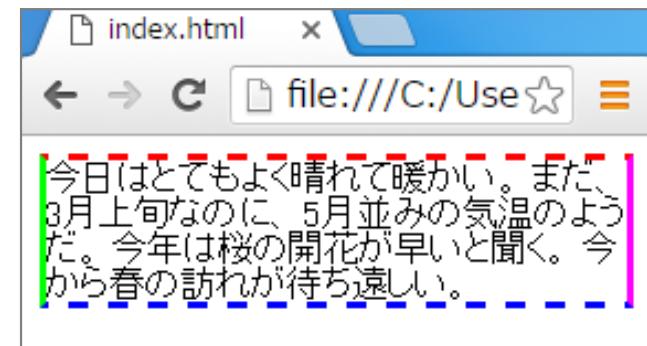
```
p {  
    border-top: 3px dotted #0F0;  
    border-bottom: 3px dashed #F0F;  
    border-left: 5px solid #F00;  
    border-right: 3px double #00F;  
}
```



# 個別に設定する（線幅、線種、色）

- サブプロパティとして、**-width**, **-style**, **-color** を付けることで、線幅ごと、線種ごと、色ごとに設定することができる

```
p {  
    border-width: 3px;  
    border-style: dashed solid;  
    border-color: #F00 #F0F #00F #0F0;  
}
```



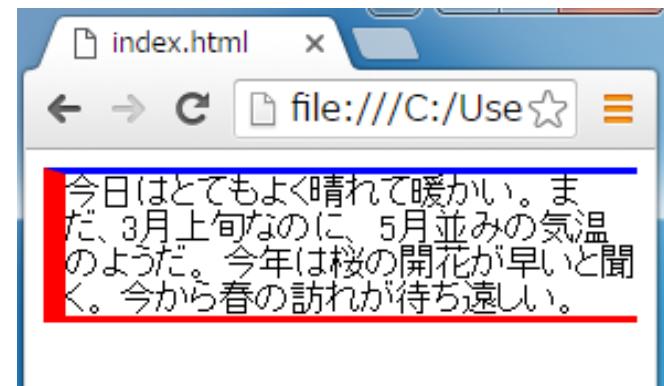
- 値は半角アキで複数指定できる

指定数	例	意味
1	<b>border-width: 3px;</b>	上下左右すべて 3px
2	<b>border-width: 1px 5px;</b>	上下は1px、左右は5px
3	<b>border-style: solid dashed dotted;</b>	上は実線、左右は破線、下は点線
4	<b>border-color: #F00 #F0F #00F #0F0;</b>	上は赤、右は紫、下は青、左は緑 時計回り

# さらに個別に設定する

- top, -bottom, -left, -right に -width, -style, -color を付け加えることで、さらに個別に指定できる

```
p {  
    border: 3px solid #F00; } 基本の設定  
    border-left-width: 10px;  
    border-right-style: none;  
    border-top-color: #00F;  
} 個別の設定
```



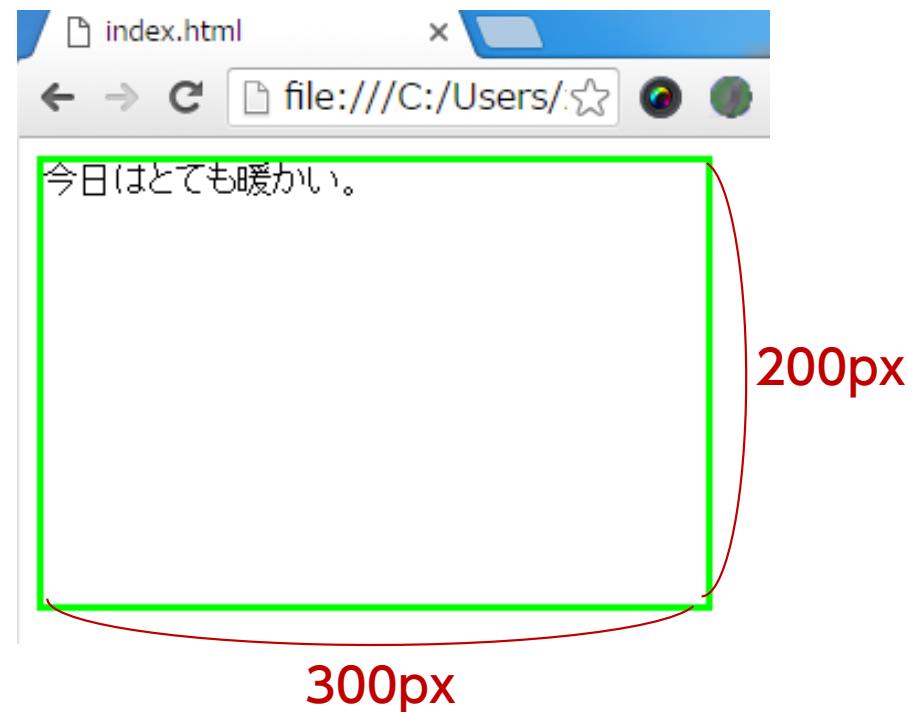
基本の設定を個別の設定で上書きしている（記述の順番が大切）

# 内容領域

- 文字や画像(img要素)が配置される領域
- 境界線や背景色を設定することで可視化できる
- 任意の幅(width)や高さ(height)を指定することができる

```
<p>今日はとても暖かい。</p>
```

```
p {  
    border: 3px solid #0F0;  
    width: 300px;  
    height: 200px;  
}
```



# 幅の設定 width

- pxや%などでの指定が可能
- デフォルト値は100%で、親要素いっぱいに広がる

```
<div>
  <p>今日はとても暖かい。</p>
</div>
```

```
div {
  border: 3px solid #00F;
  width: 300px;
  height: 200px;
}
p {
  border: 3px solid #0F0;
```

p要素には任意の幅を設定していないので、親であるdivの幅いっぱいに広がっている

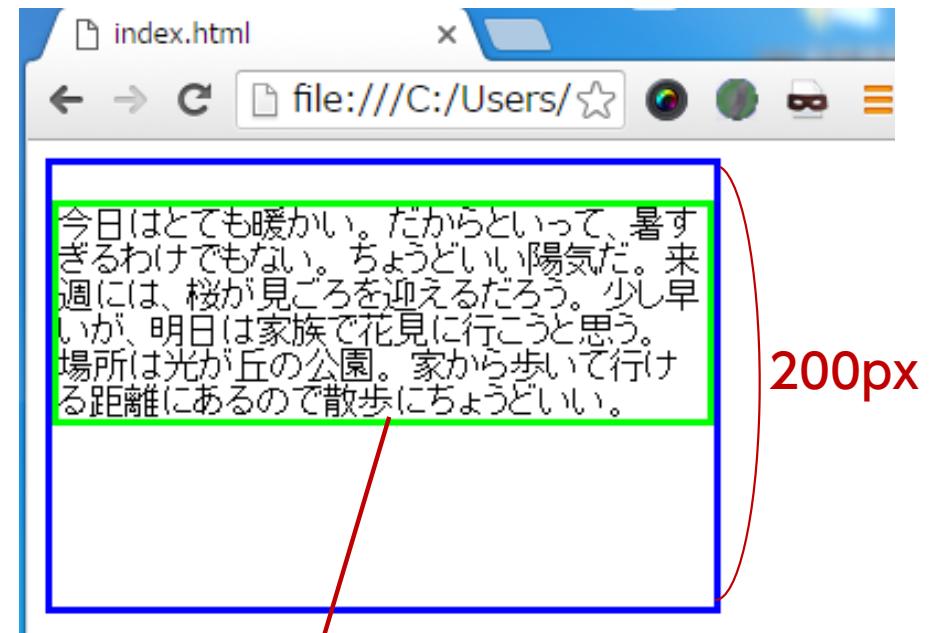


# 高さの設定 height

- pxや%などでの指定が可能
- デフォルト値はautoで、内容物の量に応じた高さになる

```
<div>
  <p>今日はとても暖かい。 (...省略)
    家から歩いて行ける距離にあるの
    で散歩にちょうどいい。
  </p>
</div>
```

```
div {
  border: 3px solid #00F;
  width: 300px;
  height: 200px;
}
p {
  border: 3px solid #0F0;
}
```



# 最大値・最小値 max-/min-

- max-widthで最大幅を、min-widthで最小幅を指定する
- max-heightで最大高を、min-heightで最小高を指定する

```
div {  
    border: 3px solid #00F;  
    width: 50%;  
    max-width: 400px;  
    min-width: 150px;  
    max-height: 100px;  
    min-height: 80px;  
}
```

400px(最大幅)

150px(最小幅)

100px  
(最大高)

50%

50%

あいうえおかきくけこさしすせそたちつてとなにつねのはひふへほまみむめもやいゆえよわいいうえを

80px  
(最小高)

50%

50%

# min(), max(), clamp()

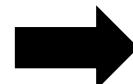
- `min()`関数は最小値を、`max()`関数は最大値を返す

```
width: 75%;  
max-width: 800px;
```



```
width: min(75%, 800px);
```

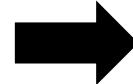
```
width: 75%;  
min-width: 380px;
```



```
width: max(75%, 380px);
```

- `clamp()`関数を使うことで、最小値、デフォルト値、最大値をまとめて設定することができる

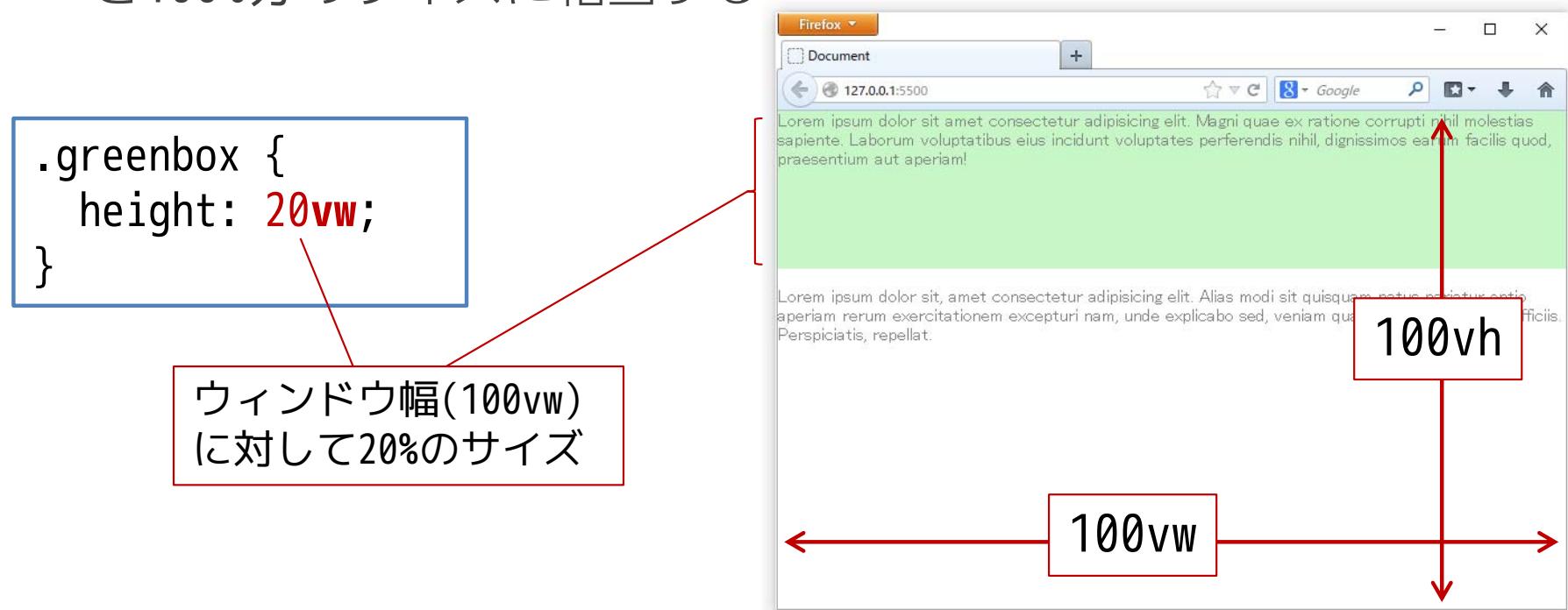
```
width: 75%;  
min-width: 380px;  
max-width: 800px
```



```
width: clamp(380px, 75%, 800px);
```

# vw, vh

- ビューポート(ブラウザウィンドウ)の幅や高さを利用する場合は、vw(Viewport Width)やvh(Viewport Height)という単位を記述する
  - 100vwがウィンドウ幅100%分のサイズ、100vhがウィンドウの高さ100%分のサイズに相当する



# vmin, vmax

- vwとvh(ウィンドウの幅と高さ)を比較し、小さい方を基準にしたい場合はvmin、大きい方を基準にしたい場合はvmaxという単位を使うことができる

```
.greenbox {  
    height: 20vmax;  
}
```

100vwと100vhを比較すると  
100vwの方が長い  
↓  
100vw(ウィンドウ幅)に対して20%のサイズ

Firefox Document 127.0.0.1:5500

100vh

100vw

# スマートフォンの画面サイズ

- スマートフォンのブラウザでは、画面をスクロールすることで、アドレスバーの表示／非表示が切り替わる
- このアドレスバーの領域を考慮した幅や高さの単位として、以下のようなものが存在する

単位	説明
svw / svh	アドレスバーの領域を差し引いた幅や高さ
lvw / lvh	アドレスバーの領域を差し引かない幅や高さ ⇒ ブラウザ全体の幅や高さ
dvw / dvh	アドレスバーの表示／非表示の変化に応じた幅や高さ

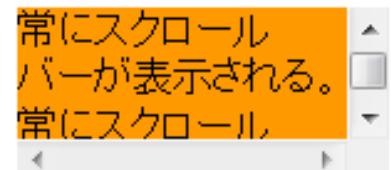
# overflowプロパティ

- コンテンツがボックス内に収まりきらない場合の表示に関する指定

値	説明
visible	はみ出した部分も表示される(初期値)
hidden	はみ出した部分は表示されない
scroll	常にスクロールバーが表示され、はみ出さない。scroll-x, scroll-yという値も使用可能
auto	文字量がボックス領域を超えた時、スクロールバーを表示

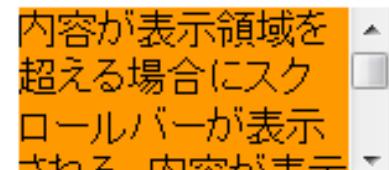
## visible

はみ出した部分も表示される。はみ出した部分も表示される。はみ出した部分も表示される。

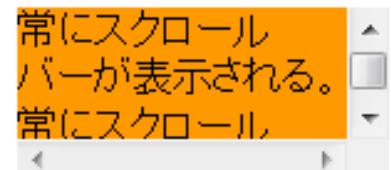


## hidden

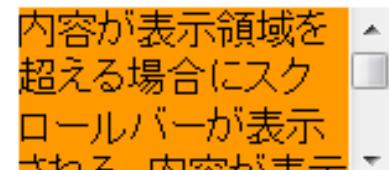
はみ出した部分は表示されない。はみ出した部分は表示されない。はみ出した部分は表示されない。



## scroll



## auto



# 練習

---

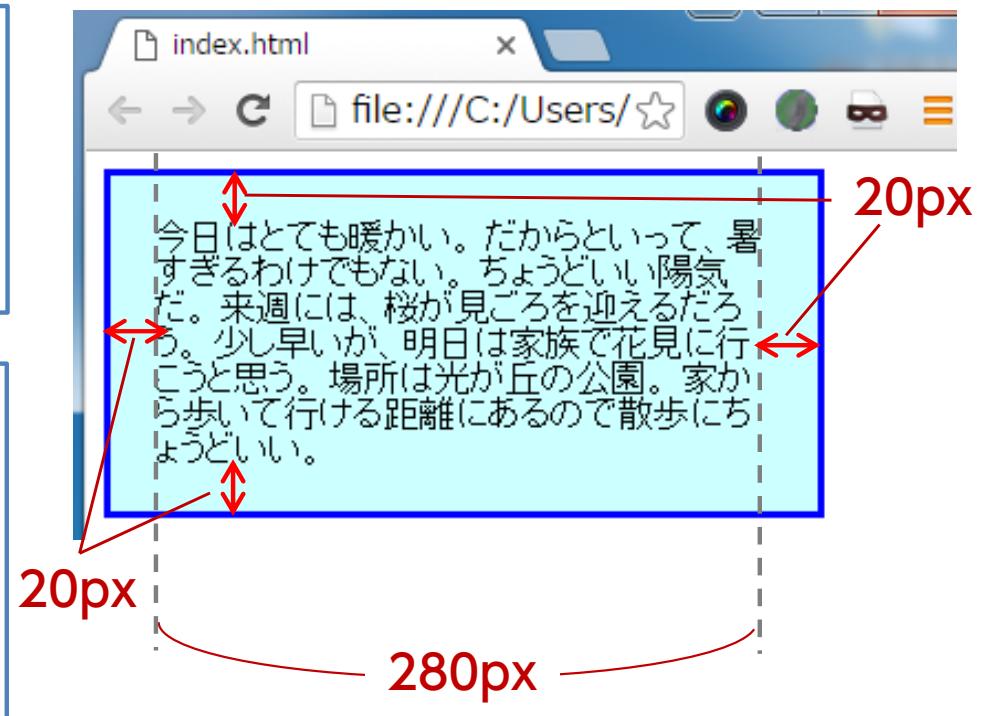
- 練習09-2

# 内側の余白 padding

- 境界線と内容領域の間の余白の指定
- pxや%などでの指定が可能
- 背景色／背景画像は、この余白の部分にも表示される

```
<p>  
今日はとても暖かい。 (...省略) 家から歩いて行ける距離にあるので散歩にちょうどいい。  
</p>
```

```
p {  
    background-color: #CFF;  
    border: 3px solid #00F;  
    width: 280px;  
    padding: 20px;  
}
```



# padding の個別指定

- ・ ボーダーと同様の方法で上下左右に個別の指定が可能
- ① サブプロパティに **top**, **bottom**, **left**, **right** を付ける

```
p {  
  padding-top: 20px;  
  padding-bottom: 20px;  
  padding-left: 10px;  
  padding-right: 0px;  
}
```

- ② 半角アキで複数の値を入れる

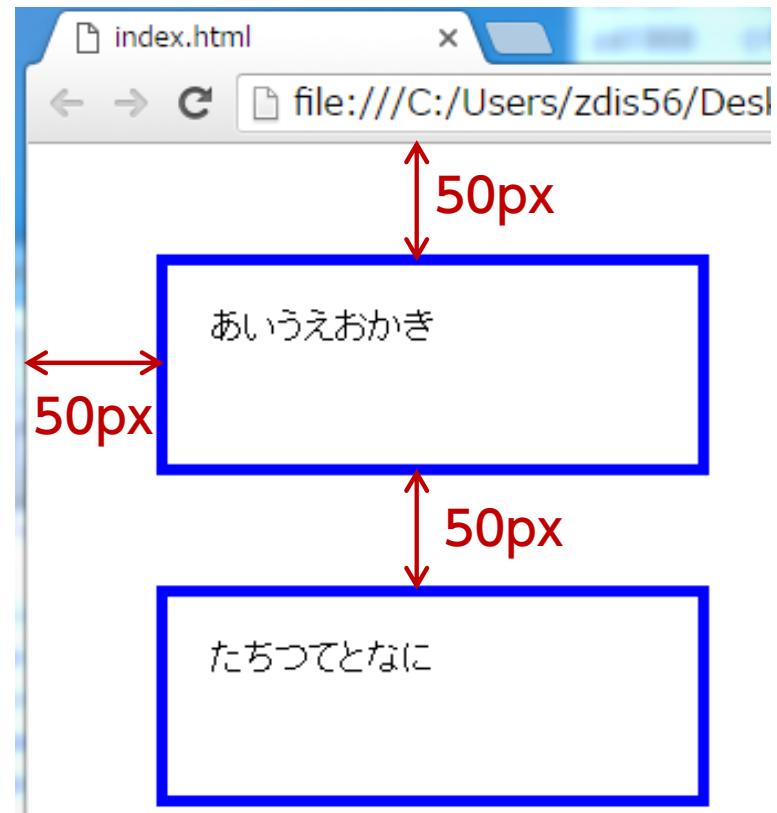
指定数	例	意味
1	<b>padding: 5px;</b>	上下左右すべて 5px
2	<b>padding: 10px 5px;</b>	上下は10px、左右は5px
3	<b>padding: 10px 0px 20px;</b>	上は10px、左右は0px、下は20px
4	<b>padding: 10px 0px 10px 5px;</b>	上は10px、右は0px、下は10px、左は5px 時計回り

# ボックスの間隔 margin

- 隣接するボックスからの距離の指定
- pxや%などでの指定が可能

```
<p>あいうえおかき</p>
<p>たちつてとなに</p>
```

```
p {
  width: 200px;
  height: 50px;
  border: 5px solid #00F;
  padding: 20px;
  margin: 50px;
}
```



# margin の個別指定

- ・ パディングと同様の方法で上下左右に個別の指定が可能
- ① サブプロパティとして-top, -bottom, -left, -rightを付ける

```
p {  
    margin-top: 20px;  
    margin-bottom: 20px;  
    margin-left: 10px;  
    margin-right: 0px;  
}
```

- ② 半角アキで複数の値を入れる

指定数	例	意味
1	<b>margin: 5px;</b>	上下左右すべて 5px
2	<b>margin: 10px 5px;</b>	上下は10px、左右は5px
3	<b>margin: 10px 0px 20px;</b>	上は10px、左右は0px、下は20px
4	<b>margin: 10px 0px 10px 5px;</b>	上は10px、右は0px、下は10px、左は5px 時計回り

# padding と margin

- 内側の余白は、親ボックスの padding と子ボックスの margin の2通りの選択肢がある

paddingで設定する場合

```
.parent {  
    background: #CF9;  
    box-sizing: border-box;  
    width: 300px;  
    height: 200px;  
    padding: 30px;  
}  
.child {  
    background: #0A0;  
    width: 100px;  
    height: 100px;  
}
```

```
<div class="parent">  
    <div class="child"></div>  
</div>
```



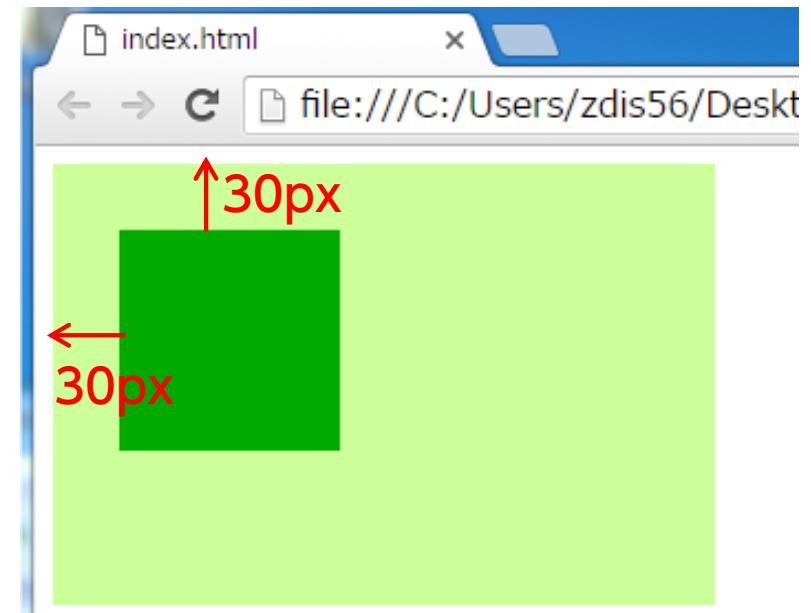
# padding と margin

## marginで設定する場合

```
.parent {  
    background: #CF9;  
    overflow: hidden;  
    width: 300px;  
    height: 200px;  
}  
  
.child {  
    background: #0A0;  
    width: 100px;  
    height: 100px;  
    margin: 30px;  
}
```

overflow:hiddenを入れないと、上のマージンが効かない（または、.parent内にテキストを入れる必要がある）

```
<div class="parent">  
    <div class="child"></div>  
</div>
```



# 上下のマージン

- 上下でマージンが重なった場合、大きい値が優先される

```
<p class="p1">あいうえおかき</p>
<p class="p2">たちつてとなに</p>
```

```
p {
    border: 1px solid #333;
}
.p1 {
    margin-bottom: 50px;
}
.p2 {
    margin-top: 20px;
}
```

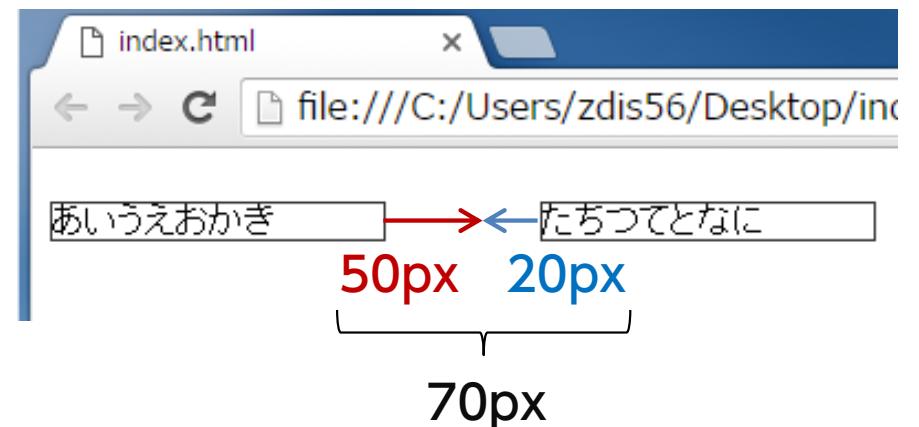


# 左右のマージン

- 左右でマージンが重なった場合、加算される

```
<p class="p1">あいうえおかき</p>
<p class="p2">たちつてとなに</p>
```

```
p {
    border: 1px solid #333;
    width: 150px;
    float: left;
}
.p1 {
    margin-right: 50px;
}
.p2 {
    margin-left: 20px;
}
```



# inline-blockのマージン

- displayプロパティがinline-blockの要素では、上下であってもマージンは加算される



```




```

```
img {
    border: 2px solid #CCC;
    padding: 2px;
    margin: 25px;
}
```

# ボックスのサイズ

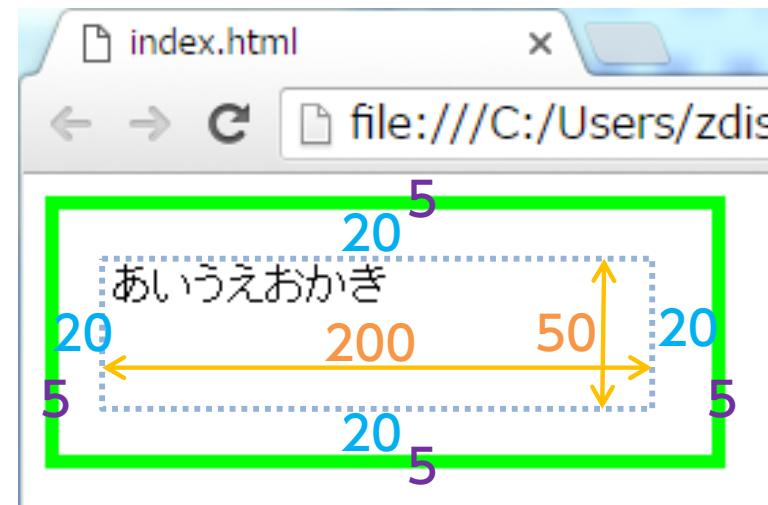
- CSSによるレイアウトはボックスを配置する作業  
⇒ ボックスのサイズを把握する必要がある

```
<p>あいうえおかき</p>
```

```
p {  
    width: 200px;  
    height: 50px;  
    padding: 20px;  
    border: 5px solid #0F0;  
}
```

ボックスの幅 = 250px  
(200 + 20 + 20 + 5 + 5)

ボックスの高さ = 100px  
(50 + 20 + 20 + 5 + 5)



widthとheightを見ると 200 × 50だが、  
ボックスとしてのサイズは 250 × 100 になる

# ボックスのサイズ調整

- 必要に応じて、width / height からパディングやボーダーの大きさを差し引く

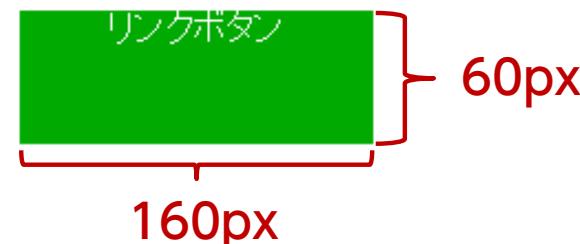
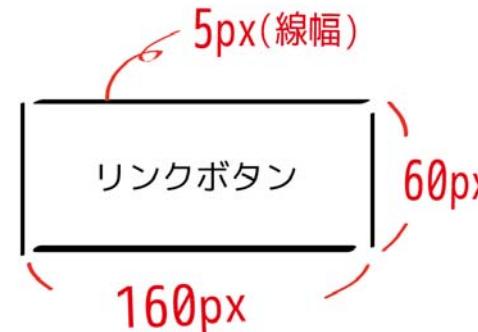
リンクボタンを作る場合の例

① 作るものを作る

② widthとheightを設定する

```
<a href="">リンクボタン</a>
```

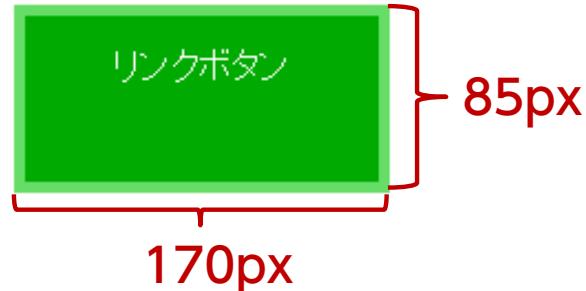
```
a {  
    color: #FFF;  
    text-decoration: none;  
    background: #0A0;  
    display: block;  
    text-align: center;  
    width: 160px;  
    height: 60px;  
}
```



# ボックスのサイズ調整

## ③ パディングとボーダーを設定

```
a {  
  ... 省略 ...  
  width: 160px;  
  height: 60px;  
  border: 5px solid #6D6;  
  padding-top: 15px;  
}
```



パディングとボーダーの分、  
サイズが大きくなる

## ④ パディングとボーダーで増えた分をwidthとheightから差し引く

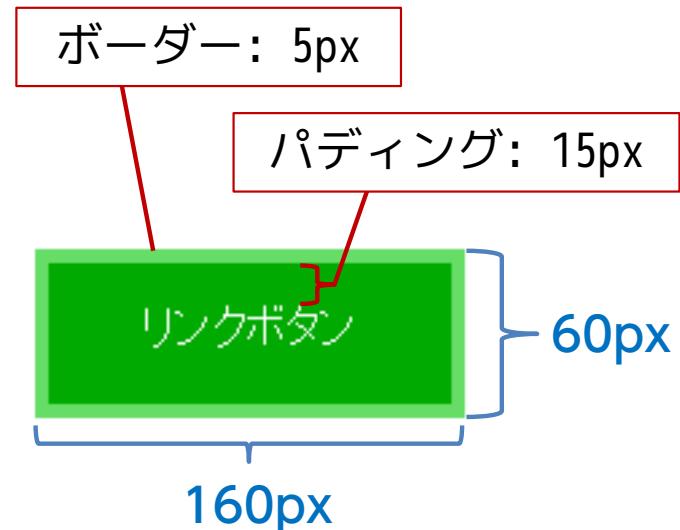
```
a {  
  ... 省略 ...  
  width: 150px;  
  height: 35px;  
  border: 5px solid #6D6;  
  padding-top: 15px;  
}
```



# calc()によるサイズ調整

- calc()関数は、引数として受け取った計算式の結果を返す
    - 単位付きの計算式を受け取ることができる
- ④' パディングとボーダーで増えた分をcalc()関数を使い、差し引く

```
a {  
  ... 省略 ...  
  width: calc(160px - (5px * 2));  
  height: calc(60px - (5px * 2) - 15px);  
  border: 5px solid #6D6;  
  padding-top: 15px;  
}
```



# box-sizing

- box-sizingプロパティを利用してすることで、widthとheightが、そのままボックスの幅と高さになる

```
<p>あいうえおかき</p>
```



The screenshot shows a browser window with the title "index.html". The address bar shows "file:///C:/Users/". Inside the window, there is a single paragraph element containing the text "あいうえおかき". The paragraph has a green border representing the content box, which is labeled "100px" on its right side. A red border representing the total box (content + padding + border) is labeled "200px" at the bottom. A black bracket on the left points from the text "box-sizing: border-box;" in the CSS to the green content box.

初期値は content-box

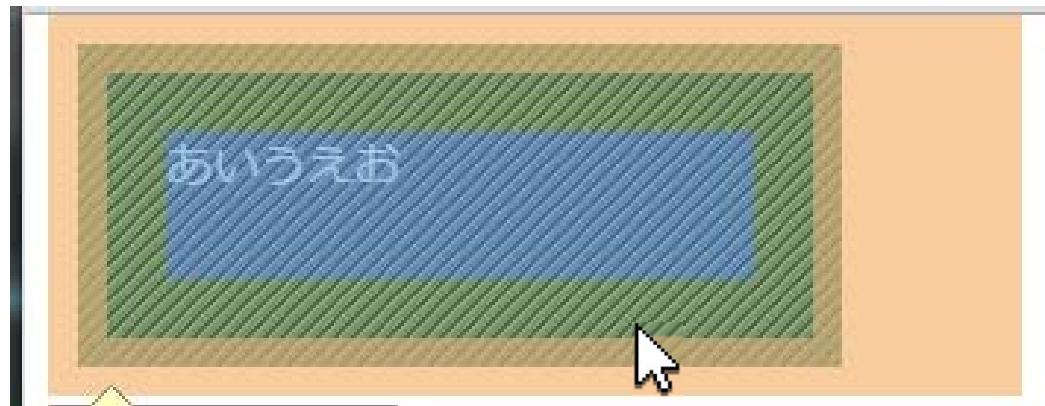
# displayプロパティとボックス

- displayプロパティによってボックスの性質は異なる

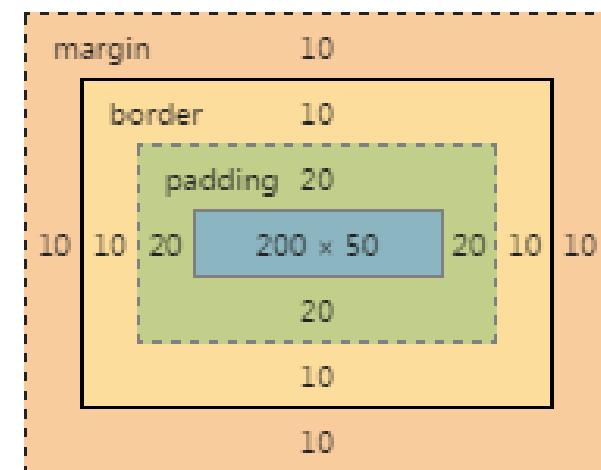
CSSプロパティ	block	inline
height (高さ)	指定可能	指定不可
width (幅)	指定可能	指定不可
padding (上下)	指定可能	指定可能 ※ボーダーの位置と背景色に影響が出るが、レイアウトには変化なし
padding (左右)	指定可能	指定可能
border (上下)	指定可能	指定可能 ※行ごとの上下につく
border (左右)	指定可能	指定可能 ※最初の行の左と最後の行の右につく
margin (上下)	指定可能	指定不可
margin (左右)	指定可能	指定可能

# 開発者用ツールの利用

- 開発者用のツールを利用してボックスモデルをブラウザ上で確認することができる



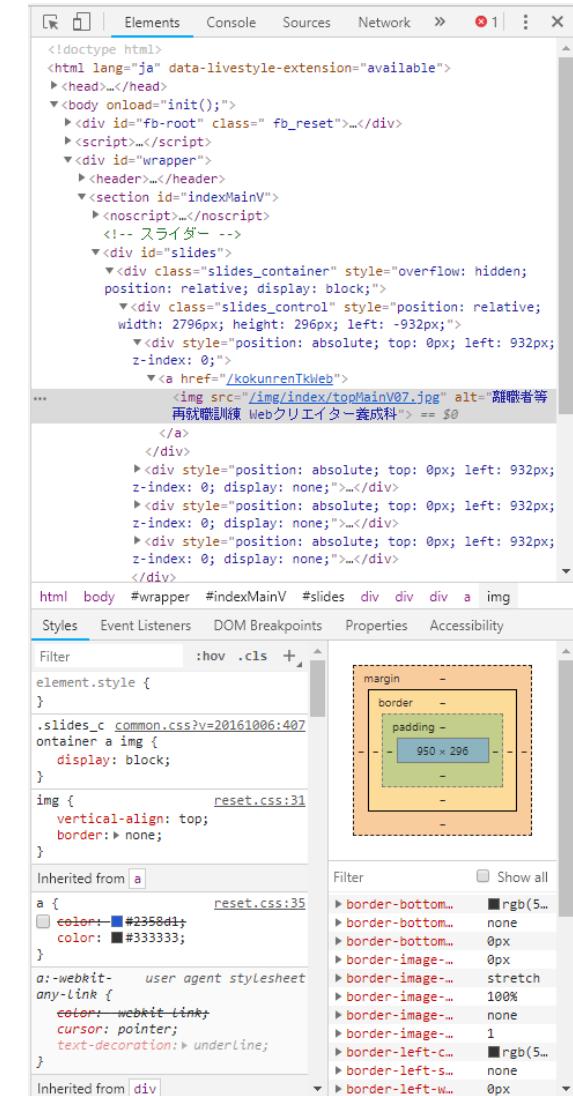
レイアウトがうまく反映されないとき  
このツールで余白やプロパティ、クラス  
名などを検証し、問題点を見つけ出す



ボックスをクリック  
すると設定した数値  
が右下に表示される

# 開発者用ツールの利用

- ディベロッパー(開発者)ツールを利用することで、ボックスの構造を観察的に把握したり、JavaScriptのエラー箇所の確認などが行える
- Chromeをはじめ、各ブラウザには開発者用のツールが備わっている
  - Chromeの場合、F12キーで右図のような画面が起動する  
⇒ ページ内の特定の場所について調べたい場合は、**右クリック»検証** または **Ctrl + Shift + C** のツールを利用する



# 練習

---

- 練習09-3

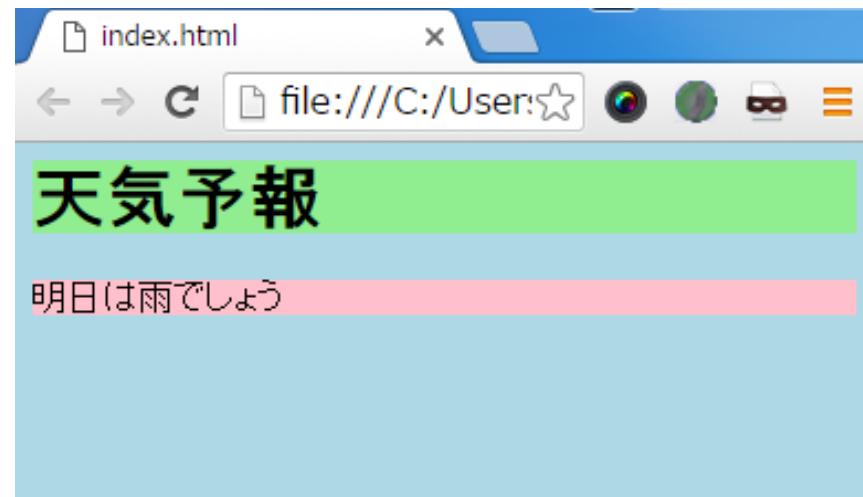
# 背景の設定

# 背景色 background-color

- 背景の色を指定する。色の指定方法は文字の色と同様

```
<body>
<h1>天気予報</h1>
<p>明日は雨でしょう</p>
</body>
```

```
body {
  background-color: lightblue;
}
h1 {
  background-color: lightgreen;
}
p {
  background-color: pink;
}
```



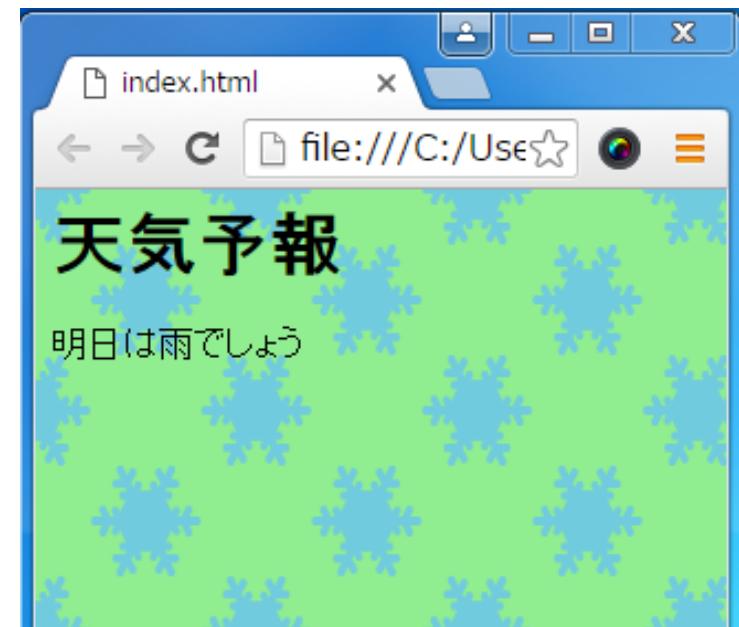
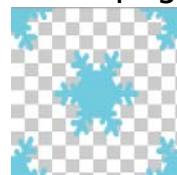
# 背景画像 background-image

- 背景画像を指定する
- 値として **url("画像のパス")** を指定する
- 背景画像は背景色よりも前面に表示される

```
<body>
<h1>天気予報</h1>
<p>明日は雪でしょう</p>
</body>
```

```
body {
  background-color: lightgreen;
  background-image: url("../images/snow.png");
}
```

snow.png



# 繰り返し background-repeat

- 背景画像の繰り返しの有無、方向を指定する

値	説明
<b>no-repeat</b>	繰り返さない
<b>repeat-x</b>	X軸方向（横方向）に繰り返す
<b>repeat-y</b>	Y軸方向（縦方向）に繰り返す
<b>repeat</b>	縦・横両方向に繰り返す（初期値）



no-repeat



repeat-x



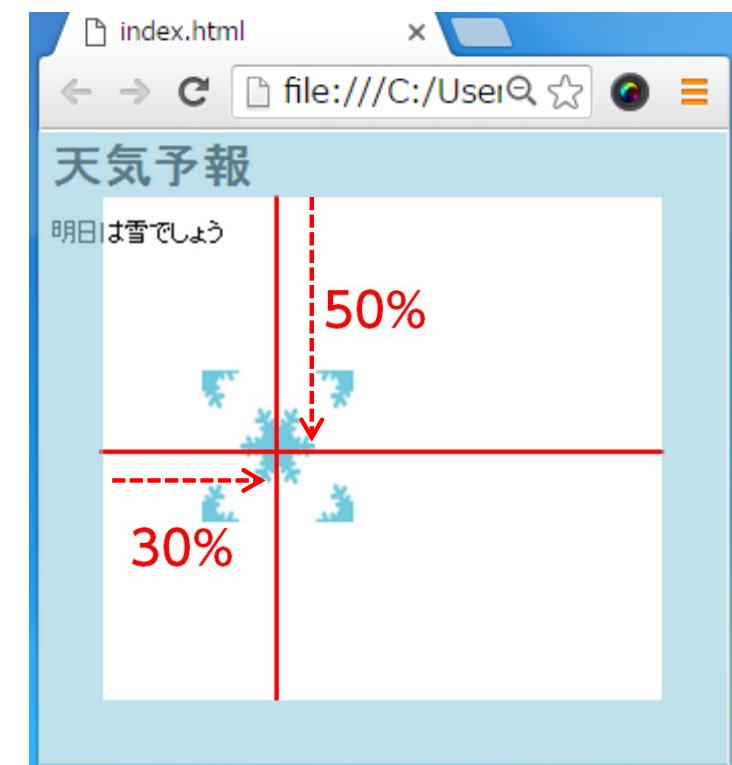
repeat-y

# 位置 background-position

- 背景画像の位置を指定する
- 値はX軸方向、Y軸方向の順で半角アキで指定する
- 値はpxや%の単位を伴う数値、または left, centerなどのキーワードで指定することができる

キーワード	説明
<b>left</b>	X軸方向 0%に相当
<b>right</b>	X軸方向 100%に相当
<b>center</b>	X,Y軸方向 50%に相当
<b>top</b>	Y軸方向 0%に相当
<b>bottom</b>	Y軸方向 100%に相当

```
body {  
    background-image:url("../images/snow.png");  
    background-repeat: no-repeat;  
    background-position: 30% 50%;  
}
```



水色部分は画像サイズに応じる

# 位置 background-position

- デフォルトでは左上が基準だが、キーワードを使うことで基準を変えることができる

左下を基準にする例

```
background-position: left 30% bottom 20px;
```

右上を基準にする例

```
background-position: right 30px top 20px;
```

右下を基準にする例

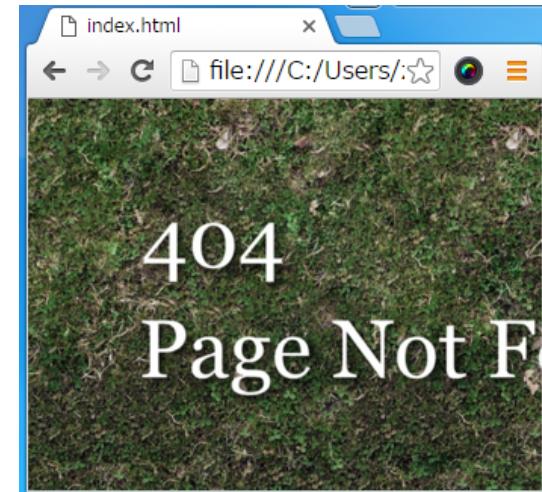
```
background-position: right 30% bottom 20%;
```

# 大きさ background-size

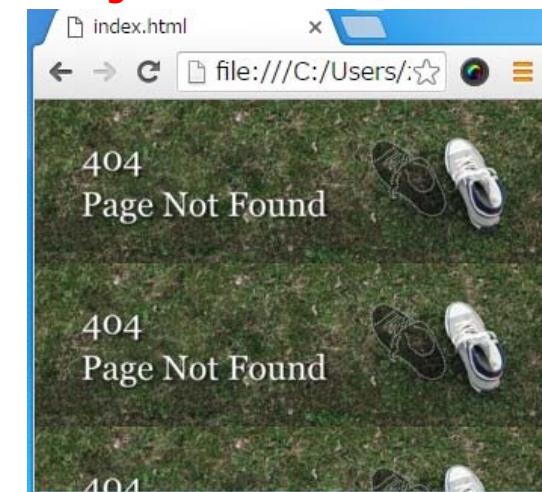
- 背景画像の大きさを指定する
- pxや%で値を指定する場合、値は幅、高さの順で半角アキで指定することが可能。値が一つだけの場合は、幅に適用され、高さは元画像の比率に応じる
- 値ではなく、以下のようなキーワードを伴う場合がある

キーワード	説明
<b>auto</b>	元画像の大きさ（に応じた比率）
<b>cover</b>	領域を一枚画像で埋め尽くす
<b>contain</b>	領域に画像のすべてが表示されるよう比率を変えずに変形する

**background-size: cover;**



**background-size: contain;**



# 背景の設定をまとめる

- **background**プロパティを利用してすることで、背景色や背景画像といった背景の設定をまとめて記述することができる

記述例

```
background: #333 url("../images/a.jpg") no-repeat center center;  
background-size: cover; /* backgroundの後に指定する */
```

※background-sizeもまとめる場合は、background-positionの後に  
/ (スラッシュ) 区切りで記述する

# 練習

---

- 練習09-4

# 補足: その他のスタイル

# ドロップシャドウ box-shadow

- text-shadowと同様の方法で、ボックスに対してドロップシャドウを設定することができる

書式

box-shadow: 横方向の位置 縦方向の位置 ぼかし 影の色;

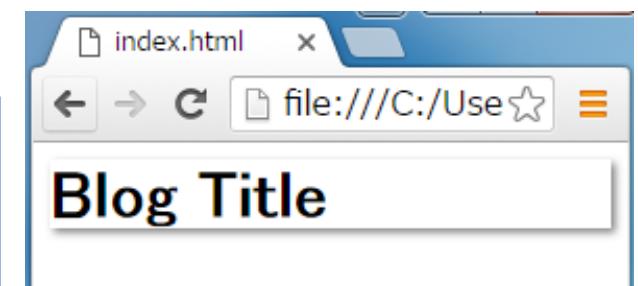
プラスの値で右、マイナスの値で左

縦方向の位置  
プラスの値で下、マイナスの値で上

記述例

```
<h1>Blog Title</h1>
```

```
h1{  
    box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.5);  
}
```



# 不透明度の指定 opacity

- 0~1の値によって不透明度を指定することができる
  - 0 : 透明 1 : 不透明

```
<div>
<h1>東京の天気</h1>
<p>今日はいい天気です</p>
</div>
```

```
div {
    background-color: #0F0;
    opacity: 0.3;
}
```



# 角丸 border-radius

- ボックスの角の丸みを指定することができる

```
div {  
    border: 1px solid #333;  
    border-radius: 3px;  
}
```



- 角ごとに、個別の設定をすることも可能

```
div {  
    border: 1px solid #333;  
    border-radius: 0 5px 10px 20px;  
}  
                                左上から時計回り
```



# 角丸 border-radius

- 角丸を応用して、円形を作ることができる

```

```

```
img {  
    border-radius: 150px;  
}
```

50%でも可



# グラフィック効果 filter

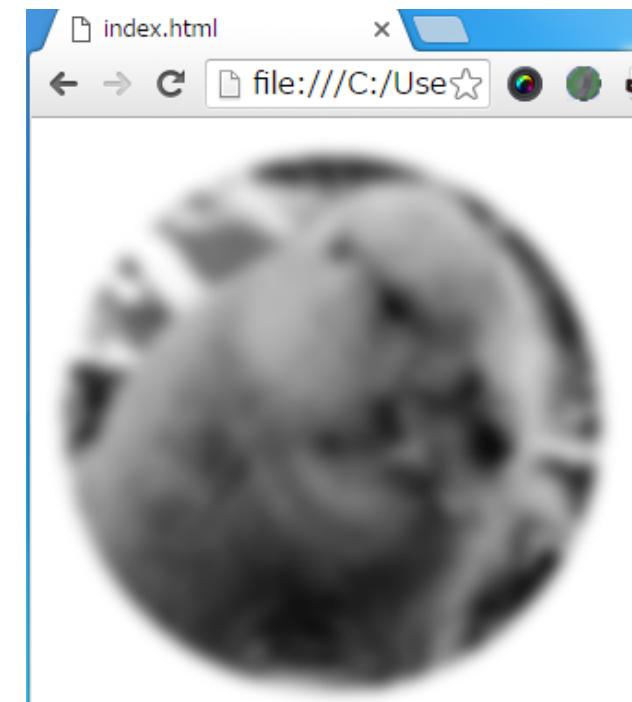
- ぼかしや色の変更などの効果を適用できる

```

```

```
img {  
    border-radius: 50%;  
    filter: blur(5px) grayscale(100%);  
}
```

他にも  
contrast(), drop-shadow(), hue-rotate()  
などの関数を記述できる



参考: <https://developer.mozilla.org/ja/docs/Web/CSS/filter>

# グラデーション

- 背景画像としてグラデーションを適用することができる
  - linear-gradient(線形)とradial-gradient(円形)がある
  - background-colorではなく、background-**image**であることに注意する

```
<div>
<h1>東京の天気</h1>
<p>今日はいい天気です</p>
</div>
```



```
div {
  background: linear-gradient(#FFF, #00F);
}
```

上側の色 下側の色

# linear-gradient

- degという形式で、角度の指定ができる
  - デフォルトは180deg

```
div {  
    background: linear-gradient(150deg, #FFF, #00F);  
}
```



- カンマ区切りで複数の値を指定することも可能

```
div {  
    background: linear-gradient(  
        #99F 0%,  
        #66F 49%,  
        #33F 50%,  
        #66F 100%);  
}
```



# radial-gradient

- 円形グラデーションは、線形グラデーションと同様の方法で指定できる

```
<div>
<h1>東京の天気</h1>
<p>今日はいい天気です</p>
</div>
```



```
div {
  background: radial-gradient(#FFF, #00F);
}
```

中心の色      外側の色