Lezione 4: CSS3

Introduzione

- CSS fondamentale per supportare il passaggio verso HTML5 dove rappresentazione della struttura e semantica sono separati da elementi stilistici

- CSS3 cambia il modo di operare
  - Specifica monolitica sostituita da approccio modulare
  - Ogni modulo tratta un tema diverso
  - Ciascun tema sviluppato con tempistiche diverse

Introduzione

- Strumenti tipografici avanzati
  - Maggiore controllo sul testo
  - Gestione avanzata della disposizione di oggetti su più colonne
  - Opacità elementi
  - Bordi con angoli smussati
- Effetti grafici
  - Transizioni
  - Trasformazioni
  - Animazioni

Introduzione

- Buona risposta da parte dei fornitori di browser
  - Firefox, Chrome, Safari, Opera supportano CSS3 fin da subito con buone percentuali
  - Explorer fornisce supporto elevatissimo dalla versione 9

Nuovi selettori/pseudoclassi

Nuovi selettori/pseudoclassi

- Nuovi selettori di attributi e pseudoclassi permettono di intervenire in maniera puntuale
  - Ad esempio, la i-esima riga <tr> di una tabella (pseudoclasse)
  - Ad esempio, posso selezionare elementi che contengono solo una parte di un valore di un attributo
- Nuovi selettori di attributi e pseudoclassi permettono di intervenire in maniera discreta
  - Discreta = non invasiva
  - Regola CSS può essere applicata con class o id senza modificare la pagina HTML
- Pulizia del codice elevata e costi di manutenzione minori
  - Maggior indipendenza tra il documento che riceve lo stile e lo stile stesso, maggior riutilizzo delle due componenti

Selettori di attributi

- Regola si applica a un elemento HTML che contiene un determinato attributo o ha un certo valore per l'attributo
  - [attribute^=value]: seleziona tutti gli elementi che hanno un attributo il cui valore inizia per value
    - a[href^="mailto:"] {
      background-color: yellow;
      }
  - [attribute*=value]: seleziona tutti gli elementi che hanno un attributo il cui valore contiene value
    - time[datetime*="-09-"] {
      background-color: yellow;
      }
  - [attribute$=value]: seleziona tutti gli elementi che hanno un attributo il cui valore termina con value
    - a[href$=".it"] {
      background-color: yellow;
      }
  - attribute-selector.html

Selettori di attributi

- Vantaggi
  - Modalità di selezione puntuale e discreta
- Svantaggi
  - Performance ridotte
  - Calcolare quanti e quali elementi sono coinvolti è oneroso
  - Utilizzare solo se necessari

Pseudo-classi struttura

- Regole per assegnare uno stile a un elemento in base alla posizione nel documento HTML
  - elem:root si riferisce all'elemento radice <html>
  - elem:empty si riferisce a elementi che non contengono nodi figli (testo o altri elementi)
    - p:empty {margin: 2px;}
    -
  - elem:only-child si riferisce a elementi che sono figli unici dell'elemento padre
    - em:only-child {color: yellow;}
    - blabla emphasized blabla

Pseudo-classi struttura

- Regole per assegnare uno stile a un elemento in base alla posizione nel documento HTML
  - elem:only-of-type si riferisce a elementi che sono figli unici, per tipo, dell'elemento padre
    - em:only-of-type {color: yellow;}
    - blabla emphasized blablabold
  - elem:nth-child(n) si riferisce all'n-esimo figlio di un elemento
    - li:nth-child(2n+1) {color: yellow;}
      - Tutti gli elementi dispari
    - li:nth-child(even) {color: yellow;}
    - li:nth-child(odd) {color: yellow;}

Pseudo-classi struttura

- Regole per assegnare uno stile a un elemento in base alla posizione nel documento HTML
  - elem:nth-last-child(n) uguale a elem:nth-child(n) solo che parte a contare dall'ultima occorrenza
  - elem:nth-of-type(n) uguale a elem:nth-child(n) solo che considera solo elementi di un certo tipo
    - Può essere usata per disporre immagini a destra e sinistra alternativamente
  - elem:nth-last-of-type(n) uguale a elem:nth-of-type(n) solo che parte a contare dall'ultima occorrenza

Pseudo-classi struttura

- Regole per assegnare uno stile a un elemento in base alla posizione nel documento HTML
  - elem:last-child si riferisce all'ultimo elemento di un elemento padre
    - a:last-child {color: yellow;}
  - elem:first-of-type si riferisce al primo elemento di un certo tipo all'interno di un elemento padre (equivalente nth-of-type(1))
    - a:first-of-type {color: yellow;}
  - elem:last-of-type si riferisce all'ultimo elemento di un certo tipo all'interno di un elemento padre (equivalente nth-last-of-type(1))
    - p:last-of-type {color: yellow;}

Pseudo-classi struttura

- Altre pseudoclassi
  - elem:target per riferirsi al target destinazione di un link
  - elem:enabled, elem:disabled, elem:checked si riferisce a elementi di una form che sono abilitati, disabilitati, contrassegnati
  - Elem:not selettore di esclusione
    - p:not(#abc) {color: white}
  - E~F si applica all'elemento F se dopo l'elemento E (non per forza consecutivamente), ed entrambi condividono il padre

Esempi

- Serie di esempi sulle pseudo-classi discusse nelle slide precedenti (class-selector.html)

@media rule e @container query

@media rule

- @media rule è usato per definire diverse regole di stile in base ai diversi media type/device
- In CSS2, venivano chiamati media types, in CSS3 vengono chiamate media queries
- Media queries guardano alle capacità del device e possono essere usate per controllare
  - width e height del viewport (l'area della pagina web visibile all'utente)
  - width e height del device
  - orientation (is the tablet/phone in landscape or portrait mode?)
  - resolution
  - e molto di più

CSS2 vs CSS3

- CSS2 si riferisce solo al media type

```
@media screen {
    p {
        font-family: verdana, sans-serif;
        font-size: 17px;
    }
}
```

- CSS3 considera anche le capability del device

```
@media screen and (min-width: 480px) {
    body
        background-color: lightgreen;
    }
}
```

Sintassi

- @media not|only *mediatype* and *(expressions)* {
  CSS-Code;
  }
  - La media query è vera se il media type specificato è consistente con il tipo di device su cui il documento viene mostrato e tutte le espressioni nella media query sono vere
  - Quando la media query è vera, gli stili vengono applicati seguendo il flusso di esecuzione tradizionale
  - A meno che vengano usati gli operatori *not* or *only*, media type è opzionale e se non specificato si riferisce ad *all*

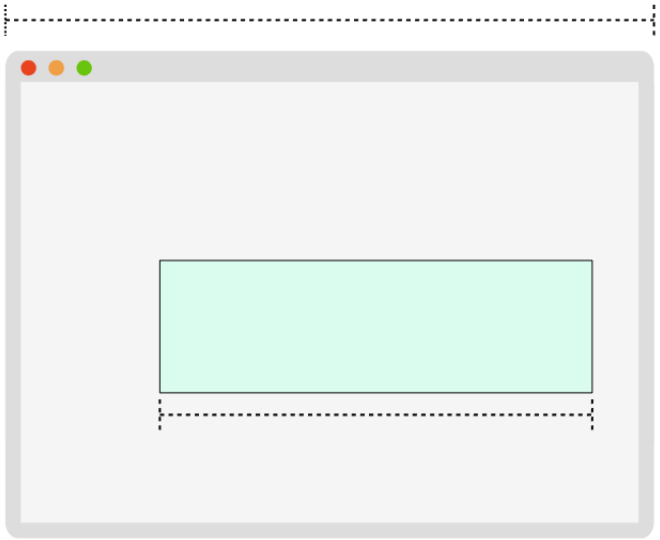
- @media screen and (min-width: 480px) {
  body {
  background-color: lightgreen;
  }
  }

Esempi

- Modifica della width porta alla modifica del colore di sfondo
  - media-color.html
- Modifica della struttura delle pagina
  - media-page.html
- Altri esempi
  - http://www.w3schools.com/css/css3_mediaqueries_ex.asp
  - http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_breakpoints

@container query

- @container query consentono di applicare stili a un elemento in base alla dimensione del contenitore dell'elemento
  - Ad esempio, se un contenitore ha meno spazio a disposizione nel contesto circostante, si possono nascondere alcuni elementi o utilizzare font più piccoli
  - Container query sono un'alternativa alle media rule, che applicano stili agli elementi in base alla dimensione del viewport o ad altre caratteristiche del dispositivo



Media query (viewport size)

```
@media (max-width: 30em){
  /* stylesheet */
}
```

Container query (container size)

```
@container (width > 15em){
  /* stylesheet */
}
```

@container query

- Per usare una container query bisogna definire un containment context
  - Proprietà container-type convalore size, inline-size, o normal
- La query nell'esempio seguente applicherà gli stili agli elementi in base alla dimensione dell'antenato più vicino con un containment context
- La query applicherà una dimensione di carattere maggiore per il titolo della card se il contenitore è più largo di 700 px
- Possibile dare un nome al containment context

```
<div class="post">
  <div class="card">
    <h2>Card title</h2>
    <p>Card content</p>
  </div>
</div>
```

```
.post {
  container-type: inline-size;
}
```

containment context

```
/* Default heading styles for the card title */
.card h2 {
  font-size: 1em;
}

/* If the container is larger than 700px */
@container (min-width: 700px) {
  .card h2 {
    font-size: 2em;
  }
}
```

containment query

Trasformazioni

Introduzione

- Un altro modulo CSS3 permette di applicare trasformazioni a un elemento HTML
  - Interviene sulle coordinate che ne determinano la posizione
- Applicazione di trasformazioni alle coordinate bidimensionali
- Applicazione di funzione di trasformazione come rotate, skew, scale, transform
  - Proprietà transform

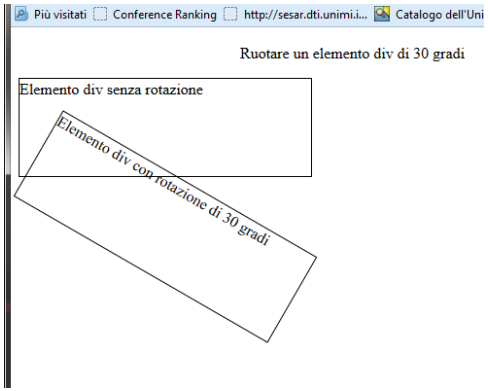
Funzione rotate()

- Ruota l'elemento in senso orario

- Specifica il numero di gradi di rotazione
  - Gradi negativi indicano rotazione in senso antiorario
  - Rotazione di 360 gradi non ha effetto sull'immagine
  - Rotazione >360 gradi viene convertita in modulo 360

- ```
  div {
      -ms-transform: rotate(90deg); /* IE 9 */
      -webkit-transform: rotate(90deg); /* Safari */
      transform: rotate(90deg);
  }
  ```

- rotate.html


  ```

Funzione scale()

- Modifica la dimensione di un elemento HTML

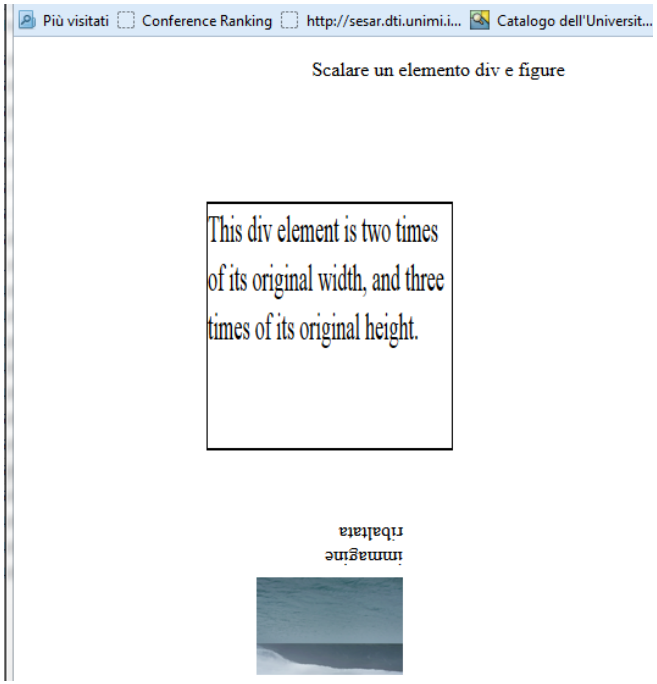
- Accetta due argomenti senza unità di misura
  - Variazione larghezza
  - Variazione altezza (opzionale)
  - Variazione <1 causa una riduzione
  - Variazione =1 mantiene le dimensioni
  - Variazione >1 causa un allargamento
  - Se il valore è negativo si applica quanto sopra e in più l'immagine è capovolta

Funzione scale()

- div {
  -ms-transform: scale(2,3); /* IE 9 */
  -webkit-transform: scale(2,3); /* Safari */
  transform: scale(2,3);
  }

- scale.html e scale2.html

- scaleX() e scaleY() modificano solo rispetto a uno degli assi X e Y
  - Cambia le proporzioni nelle figure



Funzione skew()

- Produce un'inclinazione di elemento HTML
  - È come prendere un rettangolo e tirarlo verso l'esterno da due angoli opposti fino a farlo diventare un rombo

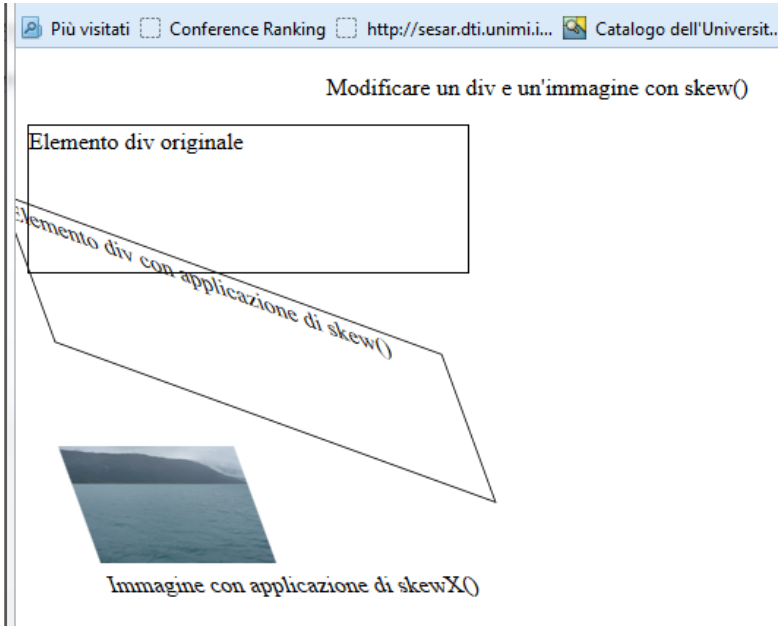
- Accetta due argomenti
  - Variazione asse orizzontale
  - Variazione asse verticale (opzionale)
  - Parametri definiscono raggi o radianti in base al quale la trasformazione sarà applicata
    - 10deg=0.17rad

Funzione skew()

- div {
  -ms-transform: skewX(20deg); /* IE 9 */
  -webkit-transform: skewX(20deg); /* Safari */
  transform: skewX(20deg);
  }

- skew.html

- skewX() e skewY() modificano solo rispetto a uno degli assi X e Y



Funzione translate()

- Sposta un elemento HTML

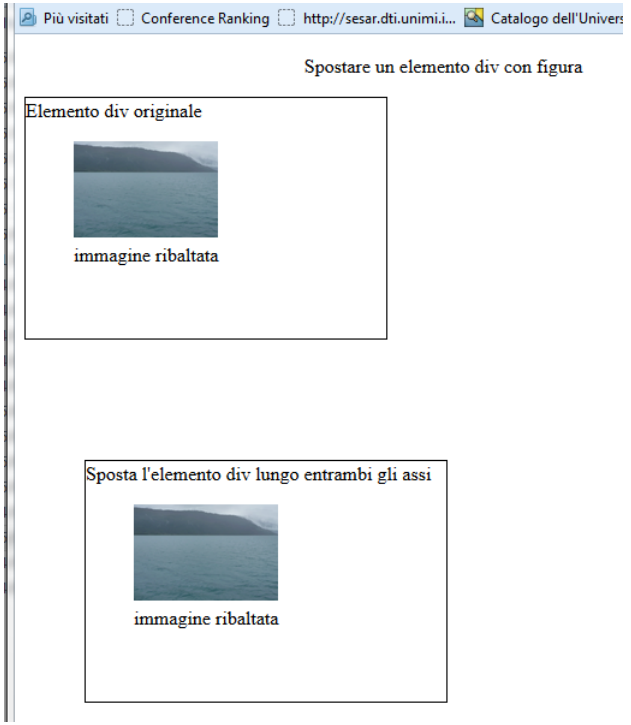
- Accetta due argomenti
  - Variazione lungo l'asse X
  - Variazione lungo l'asse Y (opzionale)
  - Parametri definiscono i pixel di spostamento

Funzione translate()

- div {
  -ms-transform: translate(50px,100px); /* IE 9 */
  -webkit-transform: translate(50px,100px); /* Safari */
  transform: translate(50px,100px);
  }

- translate.html

- translateX() e translateY() spostano rispetto a uno degli assi X e Y



Funzione matrix()

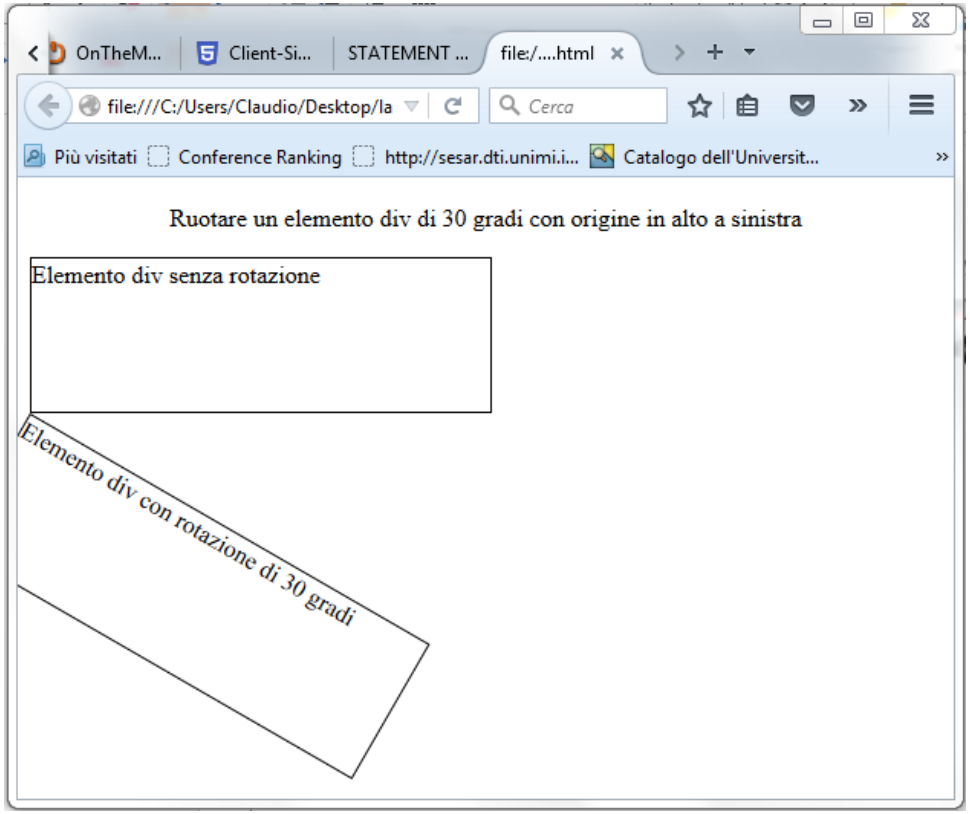
- Permette di applicare le trasformazioni tutte in un colpo
- transform: matrix(a,b,c,d,e,f);
  - I 6 parametri sono usati per applicare diverse trasformazioni
  - Rotate, scale, translate, skew
- Come alternativa si possono applicare trasformazioni multiple
  - transform: rotate(90deg) translate(50px,100px) scale(1.5)

Transform-origin

- Tutte le trasformazioni attuali fatte a partire dal centro dell'elemento
- Con transform-origin possiamo modificare questo comportamento
- Accetta due argomenti
  - Il primo argomento si riferisce al piano orizzontale
  - Il secondo argomento si riferisce al piano verticale (se non presente si usa 50%)
  - Default transform-origin:50% 50%
  - Definizione relativa (%), assoluta (em), o tramite parole chiave (top, bottom, left, right)

Transform-origin

- Riprendiamo rotate.html e applichiamo un transform-origin:top left (0em 0em) (rotate-origin.html)



Trasformazioni 3D

- Sono possibili anche trasformazioni in tre dimensioni

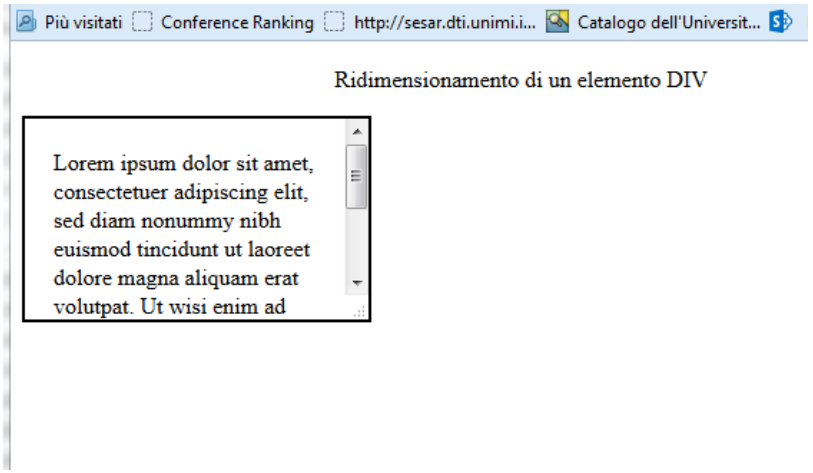
- Si prenda come esempio rotate (rotate3D.html)
  - A rotateX() e rotateY() si aggiunge rotateZ()
  - rotateZ() ruota attorno all'asse delle Z

- Esempi disponibili alla pagina http://www.w3schools.com/css/css3_3dtransforms.asp

User interface

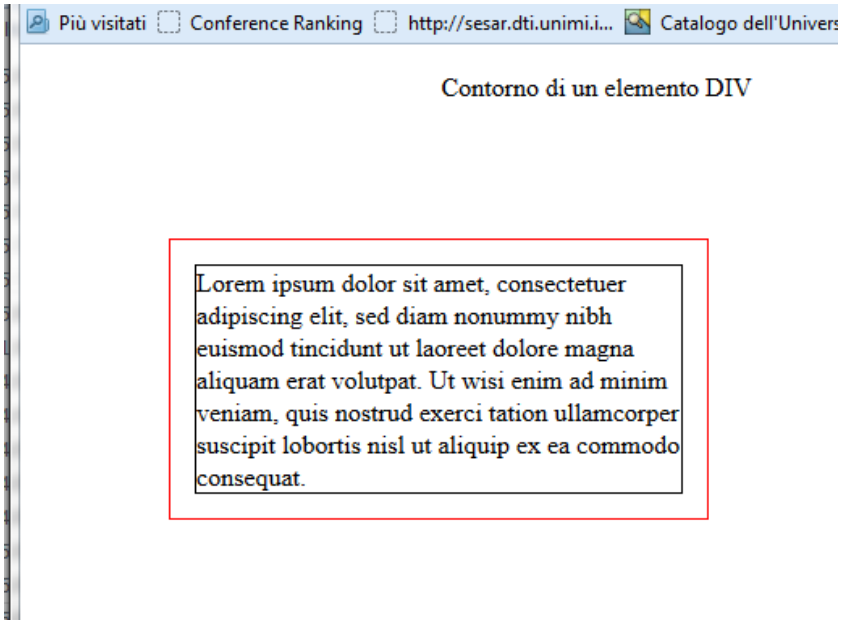
- Permette di fare ridimensionamento, contorno e dimensionamento delle box

- Ridimensionamento: proprietà resize
  - Specifica se un elemento è ridimensionabile dall'utente
  - Utile per campi di testo nelle form
  - div {
    resize: horizontal;
    overflow: auto;
    }
  - Resize può assumere anche valore vertical e both
  - resize.html



User interface

- Contorno: proprietà outline
  - Specifica un contorno attorno a un elemento
  - Specifica anche l'offset del contorno
  - Utile per evidenziare qualcosa
  - div {
    border: 1px solid black;
    outline: 1px solid red;
    outline-offset: 15px;
    }
  - outline.html



User interface

- Dimensionamento di un box: proprietà box-sizing
  - Permette di includere il padding e il border nella dimensione totale dell'oggetto a cui si riferiscono
  - Dimensione di un elemento (default)
    - width = width + padding+ border
    - height = height + padding+ border
  - Con box-sizing: border-box padding e border sono inclusi e non sommati (box-sizing.html)
  - div {
    width: 300px;
    height: 100px;
    border: 1px solid blue;
    box-sizing: border-box;
    }
    - Se al posto di div metto * si applica a tutti gli elementi HTML

Trasformazioni e animazioni

Introduzione

- Transizioni e animazioni di oggetti HTML ottenute con fogli di stile
- Grande rivoluzione
  - Prima si usavano Javascript pesanti
  - Adesso semplici CSS
- Per mantenere compatibilità con diversi browser bisognerà supportare transizioni e animazioni Javascript per un po'

Transizioni

- Abbiamo già discusso una classe di transizioni basate su eventi

- Modifica di un valore immediata
  - Ad esempio con l'evento hover
  - p {background-color: #efefef;}
    p:hover {background-color: #bbbbbb;}

- Effetto roll-over

Transizioni graduali

- Con transizioni CSS si descrive una transizione graduale da un valore iniziale a uno finale in un tempo t

- Processo di transizione
  - Identifica la proprietà target (transition-property)
  - Definisce un valore iniziale e uno finale
  - Definisce l'evento che scatena il processo di transizione (ad es., hover)
  - Definisce la durata della transizione (transition-duration, transition-timing-function, transition-delay)

Transizioni graduali

- Esempio 1: modifica della dimensione di un oggetto
  - div {
    ```
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
    transition: width 2s;
    ```
    }
  - div:hover {
    ```
    width: 300px;
    ```
    }
  - transition-div.html

Transizioni graduali

- Esempio 2: modifica del background di un oggetto
  - div {

```
width: 100px;
height: 100px;
background-color: red;
-webkit-transition: background 3s; /* For Safari 3.1 to 6.0 */
transition-property: background-color;
transition-duration:3s;
```

    }

  - div:hover {

```
background-color: blue;
```

    }

  - transition-div-background.html

Transizioni graduali

- Definizione estesa
  - Transition-property identifica la proprietà cui viene applicata la transizione
    - transition-property: background-color;
  - Transition-duration identifica l'intervallo temporale dal vecchio al nuovo valore
    - transition-duration:3s;
- Definizione compatta
  - transition: width 2s;

Transizioni graduali multiple

- Permettono di supportare scenari più complicati
- Transition-property e transition-duration accettano entrambe parametri multipli separati da virgole
- Uniamo i due esempi visti in precedenza

Transizioni graduali

- Esempio 3: modifica del background e della larghezza di un oggetto
  - div {
    ```
    width: 100px;
    height: 100px;
    background-color: red;
    -webkit-transition: background 3s; /* For Safari 3.1 to 6.0 */
    transition-property: background-color, width;
    transition-duration:3s, 2s;
    ```
    }
  - div:hover {
    ```
    background-color: blue;
    width: 300px;
    ```
    }
  - transition-div-multiplo.html

Modalità di transizione

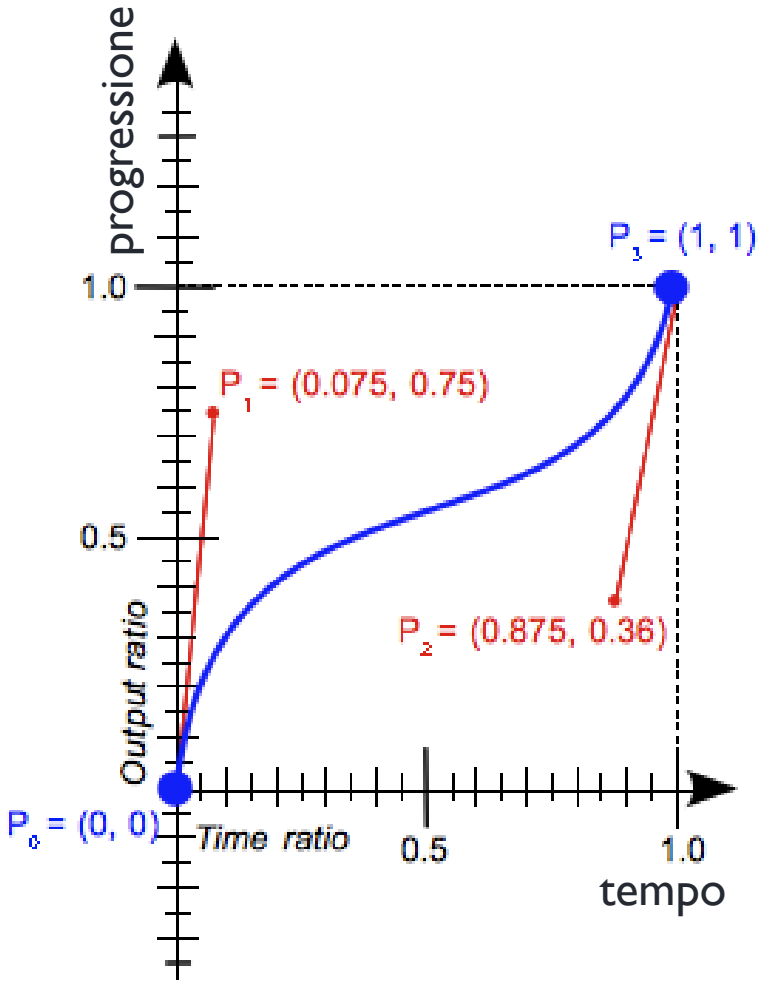
- La modalità di trasformazione nel tempo t è modificabile tramite la proprietà transition-timing-function

- transition-timing-function permette 5 valori
  - Ease (default): più ci si avvicina alla fine più si rallenta
  - Linear: transizione lineare
  - Ease-in: partenza lenta
  - Ease-out: arrivo lento
  - Ease-in-out: la somma dei due precedenti

- Riprendiamo l'esempio uno e applichiamo tutte le transizioni
  - transition-div-tfunc.html

Modalità di transizione: cubic-bezier

- Sesta modalità di trasformazione
- Cubic-bezier(x1,y1,x2,y2);
  - Definisce una curva di Bezier
  - P1=(x1,y1) rappresentano il punto di inizio dell'animazione
  - P2=(x2,y2) rappresentano il punto di fine dell'animazione
  - P0=(0,0) fisso
  - P3=(1,1) fisso
- Valori predefiniti
  - linear: cubic-bezier(0,0,1,1);
  - ease: cubic-bezier(0.25,0.1,0.25,1);
  - ease-in: cubic-bezier(0.42,0,1,1);
  - ease-out: cubic-bezier(0,0,0.58,1);
  - ease-in-out: cubic-bezier(0.42,0,0.58,1);



Modalità di transizione

- È possibile ritardare la partenza dell'elemento
- Proprietà transition-delay
  - Transition-delay:3s;
  - Ritarda la transizione di 3s
- transition-div-delay.html

Trasformazione e transizione

- Viene applicata una trasformazione alla transizione
- Ad esempio si può allargare un elemento e al tempo stesso ruotarlo (transition-transform.html)
- div {
  width: 100px;
  height: 100px;
  background: red;
  -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /* Safari */
  transition: width 2s, height 2s, transform 2s;
  }
- div:hover {
  width: 300px;
  height: 300px;
  -webkit-transform: rotate(180deg); /* Safari */
  transform: rotate(180deg);
  }

Animazioni

- Forniscono un’ulteriore semplificazione in aggiunta alle transizioni

- CSS fornisce un modulo separato per transizioni e animazioni anche se sono strettamente correlate
  - Entrambe modificano nel tempo il valore di proprietà CSS
  - Transizioni scatenate nel momento in cui i valori delle proprietà cambiano
  - Animazioni sono eseguite quando si applicano le proprietà di animazione

Animazioni: @keyframes

- Controllo di più alto livello tramite regola @keyframes

- L'animazione cambierà gradualmente gli oggetti target applicando il nuovo stile

- @keyframes <identificativo>
  {
    from, to, percentuale {
        /*una o più dichiarazioni CSS*/
    }
  }

Animazioni: @keyframes percentuale

- Identifica una serie di punti chiave attorno ai quali si svilupperà l'animazione

- Esempio di transizione multicolore
  - @keyframes example {

```
0% {background-color: red;}
30% {background-color: orange;}
60% {background-color: yellow;}
90% {background-color: green;}
100% {background-color: blue;}
```

    }
  - keyframe-percentuale.html

Animazioni: @keyframes percentuale

- Definisce una linea temporale attorno cui svolgere le cinque animazioni
- Percentuale indica la porzione di tempo dedicata a una animazione
- Per Chrome e Safari si usa la denominazione @-webkit-keyframes
- Per la prima e ultima animazione si possono usare le keyword from e to (keyframe-fromto.html)

Animazioni: @keyframes

- L'animazione da applicare a un dato elemento HTML viene riferita tramite la proprietà animation-name
- La durata dell'animazione è definita tramite la proprietà animation-duration
- Animation-name e animation-duration rappresentano il set minimo di proprietà utili per definire un'animazione
- div {

```
width: 100px;
height: 100px;
background-color: red;
-webkit-animation-name: example; /*Chrome, Safari, Opera*/
-webkit-animation-duration: 5s; /*Chrome, Safari, Opera*/
animation-name: example;
animation-duration: 5s;
```

}

Animazioni: @keyframes

- Come per le transizioni è possibile definire come avviene un passaggio da un punto all'altro
- Proprietà animation-timing-function
- Cinque valori
  - Ease (default): più ci si avvicina alla fine più si rallenta
  - Linear: transizione lineare
  - Ease-in: partenza lenta
  - Ease-out: arrivo lento
  - Ease-in-out: la somma dei due precedenti
  - keyframe-animation-timing-function.html

Animazioni: @keyframes

- Proprietà animation-iteration-count
  - Specifica quante volte l'animazione deve essere ripetuta
  - Valori da 1 a «infinite»
  - keyframe-animation-infinite.html

- Proprietà animation-delay
  - Specifica il ritardo prima di far partire l'animazione
  - 0 immediatamente, n>0 dopo n secondi, n<0 immediatamente ma l'animazione parte da un punto intermedio dipendente da n
  - keyframe-delay.html

Animazioni: @keyframes

- È possibile stabilire la direzione dell'animazione
  - Proprietà direction
  - Assume valori alternate o reverse
  - keyframe-animation-alternate.html
  - keyframe-animation-reverse.html

Animazioni: @keyframes

- Definizione compatta con proprietà animation
  - animation: <animation-name> <animation-duration> <animation-timing-function> <animation-delay> <animation-iteration-count> <animation-direction>;
  - animation: myfirst 5s linear 2s infinite alternate;

Animazioni e trasformazioni

- È possibile combinare animazioni e trasformazioni
- @keyframes example {

```
0%   {background-color:red;
                    left:0px; top:0px;}
25%  {background-color:yellow; left:200px; top:0px;
                    transform: rotate(45deg);}
50%  {background-color:blue; left:200px; top:200px;
                    transform: rotate(0deg);}
75%  {background-color:green; left:0px; top:200px;
                    transform: rotate(45deg);}
100% {background-color:red; left:0px; top:0px;
                    transform: rotate(0deg);}
}
```

- keyframe-transform.html

Web font (SOLA LETTURA)

Introduzione

- Web font permettono di estendere le opzioni possibili
- Oggi i font più utilizzati sono ancora arial, verdana, times…
  - Offrono la quasi certezza di corretta visualizzazione da parte del browser
  - body {font-family: Arial, Times, Sans-serif;}
- Risoluzione dei font in ordine di apparizione

@font-face

- La regola @font-face permette di includere qualunque tipo di carattere
  - In aggiunta ai tipi classici
- Web font supportati da tutti i browser di ultima generazione Chrome, Firefox, IE, Opera, Safari
- Supporto può essere testato da librerie come modernizr
  - Durante il caricamento della pagina o alla fine del caricamento

@font-face

- Definizione del set di caratteri
  - @font face {
    font-family: nome_set_caratteri;
    src: url(path/set_caratteri.otf);
    }
- Dopo la definizione del set di caratteri si può assegnare tale set a diversi elementi HTML
  - body {font-family: nome_set_caratteri, Sans-serif;}

Supporto formati caratteri

- Supporto dei formati di tipi caratteri
  - True type (.ttf) supportato da Chrome, Firefox, Opera, Safari
  - Open type (.otf) supportato da Chrome, Firefox, Opera, Safari
  - Embedded OpenType (.eot) supportato da IE
  - SVG font (.svg) supportato da Opera
  - Web Open Font Format (.woff) supportato da Chrome, Firefox, IE
  - Più dettagli sulle tipologie di font disponibili a http://www.w3schools.com/css/css3_fonts.asp

Supporto formati caratteri

- Evidente come la scelta del formato è critica e non basta selezionarne solo una
- Font squirrel (http://www.fontsquirrel.com/tools/webfont-generator) permette di convertire formati e molto altro…
- Tecniche cross-browser della regola @font-face
  - Scelgo un insieme di formati che mi massimizza la copertura
  - Nello scenario di mancato supporto uso generatori di caratteri tipo Cufon (https://github.com/sorccu/cufon/wiki)

Un esempio completo

- http://www.w3schools.com/css/tryit.asp?filename=trycss3_font-face_rule

```
<head>
<style>
@font-face {
  font-family: myFirstFont;
  src: url(sansation_light.woff);
}
@font-face {
  font-family: myFirstFont;
  src: url(sansation_bold.woff);
  font-weight: bold;
}
div {
  font-family: myFirstFont;
}
</style>
</head>
<body>
<div>With CSS3, websites can <b>finally</b> use fonts other than the pre-selected "web-safe" fonts.</div>
<p><b>Note:</b> Internet Explorer 8 and earlier, do not support the @font-face rule.</p>
</body>
</html>
```

Effetti tipografici: Testo ombreggiato

- Testo ombreggiato
  - Testo sfocato e parzialmente spostato
  - text-property permette di ottenere tale effetto senza dover ricorrere all'elaborazione dell'immagine
  - Applicabile direttamente al testo con meno overhead di comunicazione e pagine web più leggere
  - h1 {text-shadow: p1 p2 p3 p4;}
    - p1=scostamento orizzontale (ad es., 3px)
    - p2=scostamento verticale (ad es., 4px)
    - p3=raggio sfumatura (ad es., 2px)
    - p4=colore ombreggiatura (ad es., #0000)

Effetti tipografici: Bagliore e sfocatura

- Testo ombreggiato alla base di effetti tipografici come bagliore e sfocatura

- Bagliore
  - body {background-color: #000;}
    h1 {text-shadow: 1px 1px 5px #fff;}

- Sfocatura
  - h1 {text-shadow: 0px 0px 5px #0;}
  - Testo così a ridosso dell'ombra che sembra sfocato

- font-shadow.html

Effetti tipografici

- Ombreggiatura si può anche applicare a box
  - div {
    -mox-box-shadow: 10px 10px 5px grey; /*Mozilla*/
    box-shadow: 10px 10px 5px grey;
    }
  - I quattro parametri rappresentano: scostamento orizzontale, scostamento verticale, lunghezza del raggio di sfocatura, colore
  - Utilizzabile anche per enfatizzare un termine
    - Ad esempio, definire box-shadow per un elemento em, div
    - font-shadow-box.html
- Altri esempi
  http://www.w3schools.com/css/css3_shadows.asp

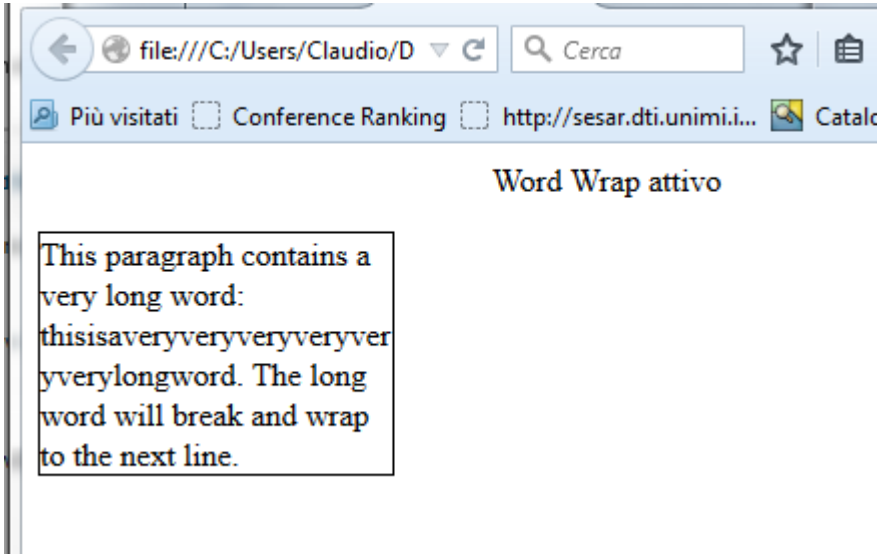
Effetti tipografici: Word wrap

- Proprietà word-wrap
  - Evita che una parola possa fuoriuscire dall'area in cui è stata confinata
  - Due possibili valori
    - word-wrap: normal; le parole non sono mai interrotte per andare a capo (valore predefinito)
    - word-wrap: break-word; le parole vengono mandate a capo quando serve

Effetti tipografici: Word wrap

- Proprietà word-wrap: evita che una parola possa fuoriuscire dall'area in cui è stata confinata
- word_wrap_yes.html

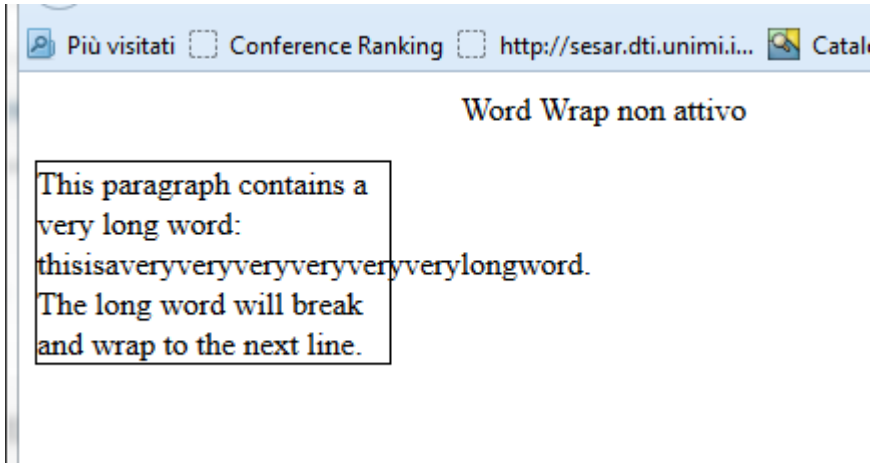
```
<html>
<head>
<style>
p.test {
   width: 11em;
   border: 1px solid #000000;
   word-wrap: break-word;
}
</style>
</head>
<body>
<p class="test"> This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.</p>
</body>
</html>
```



Effetti tipografici: Word wrap

- Proprietà word-wrap: evita che una parola possa fuoriuscire dall'area in cui è stata confinata
- word_wrap_no.html

```
<html>
<head>
<style>
p.test {
   width: 11em;
   border: 1px solid #000000;
   word-wrap: break-word;
}
</style>
</head>
<body>
<p class="test"> This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.</p>
</body>
</html>
```



Effetti tipografici: text-overflow

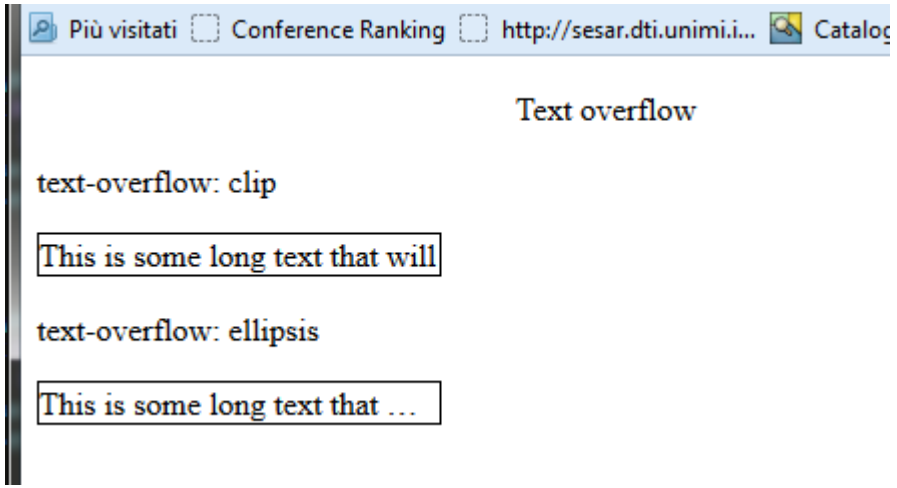
- Se il testo è troppo lungo si possono applicare dei puntini di sospensione quando il testo esaurisce lo spazio
  - p.test1 {
    white-space: nowrap;
    width: 200px;
    border: 1px solid #000000;
    overflow: hidden;
    text-overflow: clip;
    }
  - Proprietà overflow impone che il testo fuori dall'elemento contenitore non sia visualizzato
  - white-space impone al testo di non andare a capo quando c'è uno spazio

Effetti tipografici: text-overflow

- text-overflow: clip
  - p.test1 {
    white-space: nowrap;
    width: 200px;
    border: 1px solid #000000;
    overflow: hidden;
    text-overflow: clip;
    }
  - text-overflow: clip
    This is some long text that will not fit in the box

Effetti tipografici: text-overflow

- text-overflow: ellipsis
  - p.test2 {
    white-space: nowrap;
    width: 200px;
    border: 1px solid #000000;
    overflow: hidden;
    text-overflow: ellipsis;
    }
  - text-overflow: ellipsis
    This is some long text that will not fit in the box



text-overflow.html

Impaginazione testo in colonne

- Impaginazione del testo in colonne è obiettivo del modulo “multicolumn layout”
- Permette di discortarsi definitivamente dall’uso delle tabelle HTML
- Vantaggi
  - Codice HTML pulito e leggero (DOM più agile, javascript semplificato)
  - Testo scorre in modo fluido da una colonna all’altra a ogni ridimensionamento del contenitore
  - Migliore rappresentazione semantica e facilità di lettura per gli screen reader

Impaginazione testo in colonne

- Proprietà column-width (column-width.html)
  - Definisce la larghezza minima delle colonne
  - <style>

```
.newspaper {
    -webkit-column-width: 100px; /* Chrome, Safari, Opera */
    -moz-column-width: 100px; /* Firefox */
    column-width: 100px;
}
</style>
```

  - …

Impaginazione testo in colonne

- Proprietà column-count (column-count.html)
  - Definisce il numero di colonne di testo da mostrare
  - <style>

```
.newspaper {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
</style>
```

  - …

Impaginazione testo in colonne

- Notazione contratta
  - div {columns:2;} stabilisce il numero di colonne
  - div {columns:10em;} stabilisce la larghezza minima
  - div {columns:2 10em;} stabilisce il numero di colonne e la larghezza minima

Impaginazione testo in colonne

- Proprietà column-gap (column-gap.html)
  - Definisce la spaziatura tra le colonne di testo
  - <style>

```
.newspaper {
        -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
        -moz-column-gap: 40px; /* Firefox */
        column-gap: 40px;
}
</style>
```

  - …

Impaginazione testo in colonne

- Nello spazio lasciato dalla proprietà column-gap agisce la proprietà column-rule (column-rule.html)
  - Il testo inserito dalla proprietà column-rule non incide sul gap
  - È composta da tre proprietà: column-rule-width, column-rule-style, column-rule-color
  - <style>

```
.newspaper {
        -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */
        -moz-column-rule-style: solid; /* Firefox */
        column-rule-style: solid;
        -webkit-column-rule-width: 1px; /* Chrome, Safari, Opera */
        -moz-column-rule-width: 1px; /* Firefox */
        column-rule-width: 1px;
        -webkit-column-rule-color: lightblue; /* Chrome, Safari, Opera */
        -moz-column-rule-color: lightblue; /* Firefox */
        column-rule-color: lightblue;
}
</style>
```

  - …

Impaginazione testo in colonne

- Definizione di quante colonne deve occupare un testo (column-span.html)
  - Proprietà column-spam funziona in Chrome non in Firefox
  - <style>

    h2 {

    -webkit-column-span: all; /* Chrome, Safari, Opera */

    column-span: all;

    }

    </style>
  - …
  - Se al posto di «all» metto «1» il titolo h2 finisce all'interno di una singola colonna

Bordi e sfondi (SOLA LETTURA)

Introduzione

- Un modulo CSS3 di fondamentale importanza permette di gestire bordi e sfondi
- Esistono diverse proprietà
- Molte delle proprietà sono definite a partire dalla proprietà border-radius

Border-radius

- Tramite la proprietà border-radius si definiscono gli angoli arrotondati
- Prima si utilizzava una tecnica basata su immagini e tabelle
- Vantaggi border-radius
  - Minor numero di elementi
  - Nessuna necessità di utilizzare immagini per gli angoli (4 richieste HTTP in meno)
  - Proprietà sintetica

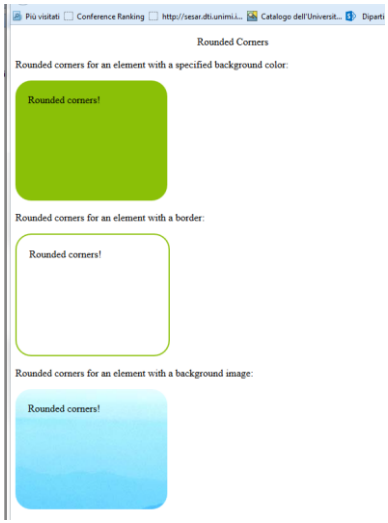
Border-radius

```
<style>
#rcorners1 {
    border-radius: 25px;
    background: #8AC007;
    padding: 20px;
    width: 200px;
    height: 150px;
}

#rcorners2 {
    border-radius: 25px;
    border: 2px solid #8AC007;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```

```
#rcorners3 {
 border-radius: 25px;
 background: url(paper.gif);
 background-position: left top;
 background-repeat: repeat;
 padding: 20px;
 width: 200px;
 height: 150px;
}
</style>
```

border-radius.html



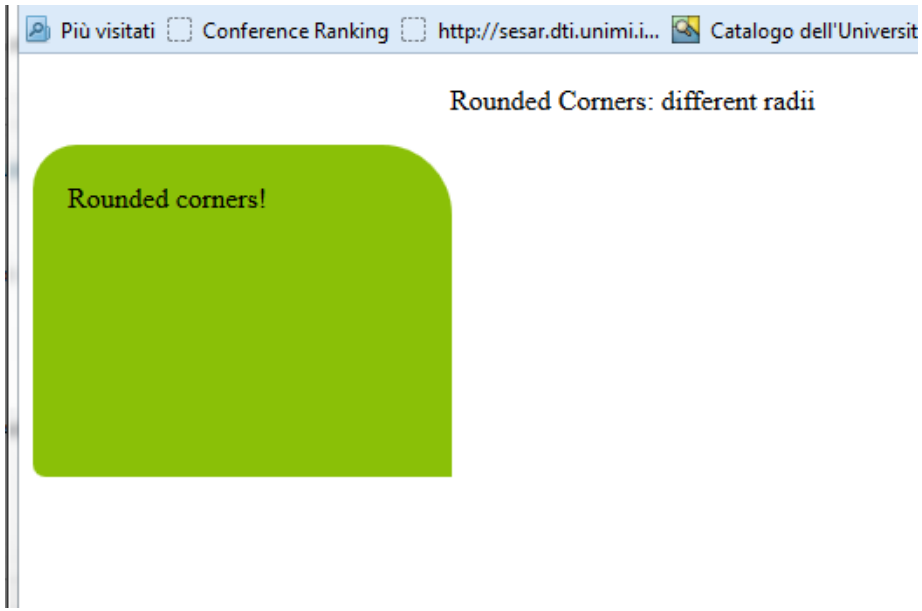
Border-radius

- Bordi definibili con un numero di parametri variabili da 1 a 4
  - Singolo valore: la stessa lunghezza del raggio è applicata a tutti gli angoli
  - Due valori: il primo valore identifica il raggio dell'angolo in alto a sinistra e in basso a destra, il secondo valore i raggi degli angoli rimanenti
  - Tre valori: il primo valore si riferisce all'angolo in alto a sinistra, l'ultimo all'angolo in basso a destra, il secondo ai rimanenti due angoli
  - Quattro valori: indicano gli angoli a partire da quello in alto a sinistra procedendo in senso orario
- Valore a zero identifica uno spigolo

Border-radius

```
<style>
#rcorners1 {
    border-radius: 25px 40px 0px 7px;
    background: #8AC007;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```

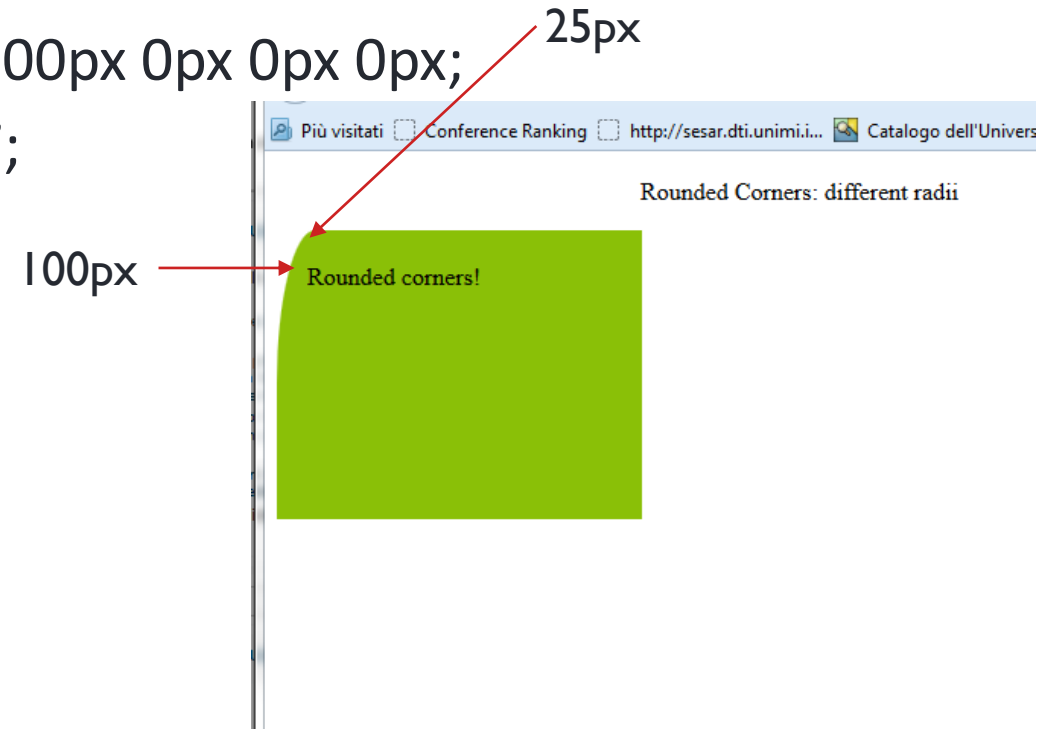
border_radius2.html



Border-radius

- Definizione di un raggio come lunghezza del raggio orizzontale e verticale: rappresentazione di una sezione di ellissi

```
<style>
#rcorners1 {
    border-radius: 25px/100px 0px 0px 0px;
    background: #8AC007;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```



border_radius3.html

Border-radius

- Meccanismi di definizione del raggio
  - Definizione assoluta: border-radius:25px/100px
  - Definizione percentuale: border-radius:10%/90%
  - Definizione estesa di border-radius:2px;
    - border-top-left-radius:2px;
      border-top-right-radius:2px;
      border-bottom-right-radius:2px;
      border-bottom-left-radius:2px;
  - Definizione estesa di border-radius:2px 10px 20px / 3px 7px;
    - border-top-left-radius:2px 3px;
      border-top-right-radius:10px 7px;
      border-bottom-right-radius:20px 3px;
      border-bottom-left-radius:10px 7px;

Border-image

- È possibile usare come bordo un'immagine
- Proprietà border-image
- Esempio: Border-image: url{border-image.png} 20 round
  - url{border-image.png}: l'immagine da usare
  - 20: distanza calcolata a partire da ciascuna estremità dell'immagine verso il centro
    - Serve per identificare la parte di immagine da utilizzare per il bordo
  - Round: indica come l'immagine deve essere usata per creare il bordo

Border-image

- Utilizzo dell'immagine (terzo parametro border-image)
  - Stretch: immagine estesa per rivestire l'area del bordo
  - Repeat: l'immagine è ripetuta per occupare tutto il bordo
  - Round: simile a repeat ma si adatta alle dimensioni del bordo

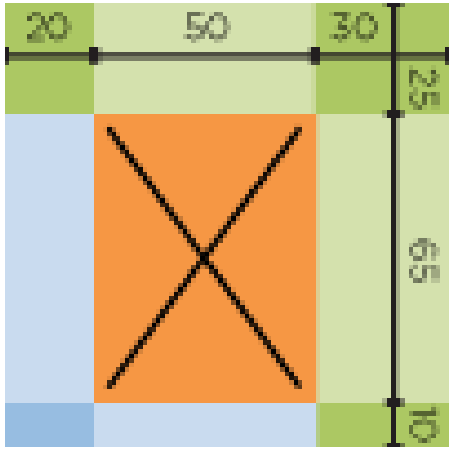
Border-image

- Versione estesa della definizione di border-image richiede di definire (border_image.html)
  - Border-image-source
    - Border-image-source: url{border-image.png}
  - Border-image-slice
    - Border-image-slice: 20
  - Border-image-repeat
    - Border-image-repeat: round

Border-image

- Border-image-slice
  - Suddivide l'immagine in 9 parti
  - Piazza gli angoli negli angoli
  - Le parti centrali vengono ripetute o estese come richiesto

- Border-image-slice: 25% 30% 10% 20%
  - Primo parametro indica top slice
  - Secondo parametro right slice
  - Terzo parametro bottom slice
  - Quarto parametro left slice



- https://css-tricks.com/understanding-border-image/

Sfondi

- Proprietà background permette di aggiungere controlli avanzati sulle immagini sfondo
  - background: url(sfondo.jpg);

- Le immagini possono essere ridimensionate (auto per le dimensioni reali)
  - Stabilisce su quale porzione dell'elemento contenitore deve essere posta l'immagine
  - Definizioni assolute
    - background-size: 480px 640px; (larghezza 480px e 640px)
  - Definizione relative
    - background-size: 36% 100%; (un terzo della larghezza e tutta l'altezza dell'elemento contenitore)
  - Contain
    - Applica la larghezza massima consentita dall'elemento contenitore mantenendo le proporzioni dell'immagine: si mantiene interno all'elemento contenitore
  - Cover
    - Dimensioni dell'immagine di sfondo occupa l'intera area dell'elemento contenitore. Preserva le proporzioni e può finire al di fuori dei limiti dell'elemento contenitore

Sfondi

- Posizione dell'immagine
  - background-position stabilisce dove l'immagine verrà posizionata
    - background-position: right bottom;
  - Proprietà aggiuntive: background-origin e background-clip (prossime slide)

- Ripetizione dell'immagine per riempire l'elemento contenitore
  - background-repeat: no-repeat, repeat
  - background_image.html

Sfondi multipli

- Sfondi multipli
  - background-image: url(img_flwr.gif), url(paper.gif);

- Definizione compatta
  - #example1 {
    background: url(img1.gif) right bottom no-repeat,
    url(img2.gif) left top repeat;
    }

- Esempio: http://www.w3schools.com/css/tryit.asp?filename=trycss3_background_multiple

Posizionamento immagine

- background-origin
  - Definisce la posizione sulla base di tre valori border-box, padding-box, content-box
  - border-box: il vertice in alto a sinistra dell'immagine coincide con il vertice sinistro esterno al bordo
    - background-origin.html
  - #example1 {
    border: 10px solid black;
    padding: 35px;
    background: url(Colline.jpg);
    background-repeat: no-repeat;
    background-origin: border-box;
    }

Posizionamento immagine

- background-origin
  - Definisce la posizione sulla base di tre valori border-box, padding-box, content-box
  - padding-box: il vertice in alto a sinistra dell'immagine coincide con il vertice sinistro esterno al padding
    - background-origin.html
  - #example1 {
    border: 10px solid black;
    padding: 35px;
    background: url(Colline.jpg);
    background-repeat: no-repeat;
    background-origin: padding-box;
    }

Posizionamento immagine

- background-origin
  - Definisce la posizione sulla base di tre valori border-box, padding-box, content-box
  - content-box: il vertice in alto a sinistra dell'immagine coincide con il vertice sinistro esterno dell'area contenuto
    - background-origin.html
  - #example1 {
    border: 10px solid black;
    padding: 35px;
    background: url(Colline.jpg);
    background-repeat: no-repeat;
    background-origin: content-box;
    }

Posizionamento immagine

- background-clip
  - Stabilisce se l'immagine si deve estendere o meno nell'area occupata dal bordo
  - Border-box (default): l'immagine si estende sopra il bordo
  - Padding-box: l'immagine non si estende sopra il bordo
  - background_clip.html

Opacità

- Definisce il grado di trasparenza
- Accetta valori tra 0 (invisibile) e 1 (visibile)
- ```
  #trasparente {
        opacity: 0.3;
        filter:alpha(opacity=30); /* For IE8 and earlier */
  }
  ```
  ```

Opacità

```
img {
    opacity: 0.4;
    filter: alpha(opacity=40); /* For IE8 and earlier */
}

img:hover {
    opacity: 1.0;
    filter: alpha(opacity=100); /* For IE8 and earlier */
}
```

Colori e gradienti

- CSS3 Colors
  - Diverse modalità di definizione RGBA, HSL, HSLA, opacity
  - Maggiori dettagli http://www.w3schools.com/css/css3_colors.asp

- CSS3 Gradients
  - Permette di mostrare transizioni tra diversi colori
  - Prima era necessario usare immagini per questi effetti, con spreco di banda e tempo
  - Miglior effetto visivo
  - Gradienti lineari (down/up/left/right/diagonally)
  - Gradienti radiali (dal centro)
  - background: linear-gradient(*direction*, *color-stop1*, *color-stop2, ...*);
  - gradient.html

Conclusioni

- CSS3
  - Specifiche modulari
  - Permettono separazione tra contenuto e visualizzazione
  - Nuovi selettori, bordi e sfondi, web font, trasformazioni, animazioni