

Textul este, cu siguranță, unul dintre cele mai importante tipuri de conținut pe care îl pot deține documentele HTML. Prima parte a acestui curs prezintă un număr mare de elemente HTML diferite ale căror rol principal a fost marcarea textului (titluri, paragrafe, liste, tabele etc.). Cu toate acestea, până în prezent, textul documentelor noastre HTML a fost redat așa cum dorea browserul. Prin urmare, această lecție va fi dedicată diferitor tehnici care ne vor permite să influențăm în mod independent aspectul și formatarea textului în documentele HTML.

## Tipografia

Înainte de a trece la diferite tehnici de stilizare a textului într-un document HTML, trebuie să abordăm principiile de bază ale tipografiei. Tipografia este, de fapt, abilitatea de formatare a textului pentru a-l prezenta într-un mod inteligibil și ușor de citit.

În trecut, în timpul apariției primelor tipografii, textul era imprimat pe hârtie folosind caractere care erau turnate din metal. Un set de astfel de caractere metalice astăzi, în forma digitală, se numește **font**.

Toate fonturile sunt împărțite în următoarele tipuri de bază:

- Serif;
- Sans Serif;
- Monospace.

Fonturile **serife** sunt caracterizate prin detalii la capetele liniilor. Aceste detalii se numesc serife, de unde provine și numele acestui tip de font. Această categorie de fonturi se numește și Roman, iar cea mai cunoscută familie de fonturi de acest tip este Times New Roman.

Fonturile **sans serif** sunt acele fonturi care se disting prin lipsa de detalii pe care le posedă fonturile serife. Acest lucru este complet logic și poate fi dedus chiar din numele acestui tip de font, dat fiind faptul că sans în franceză înseamnă „fără”. Datorită absenței așa-numitelor decorațiuni, fonturile din această categorie arată mai curat și mai modern. Cele mai populare familii de fonturi de acest tip sunt Arial, Tahoma, Verdana, Helvetica etc.

**Monospace** sunt acele fonturi la care toate caracterele ocupă același spațiu orizontal. Introducerea mașinilor de scris este, în general, considerată a fi începutul aplicării lor, când astfel de fonturi au fost necesare datorită limitărilor tehnologice. Ulterior și-au găsit aplicarea pe primele computere și terminale și sunt încă utilizate în programare. Cel mai frecvent font Monospace de pe web este Courier New (sau Courier).

Diferențele dintre cele trei categorii de fonturi enumerate sunt prezentate în imaginea 17.1.



*Imaginea 17.1. Diferențele dintre cele trei tipuri de categorii de fonturi de bază*

## Familia de fonturi

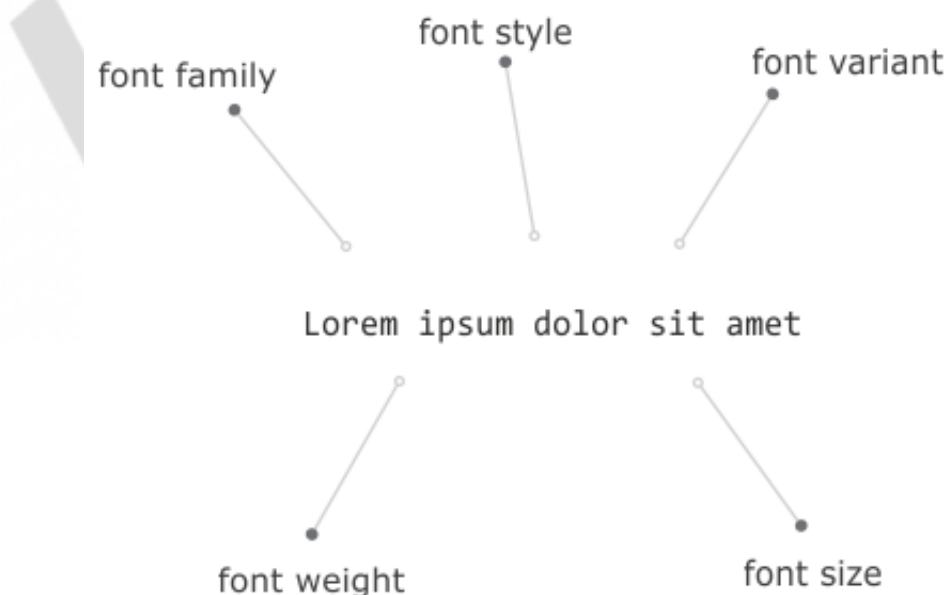
Mai multe fonturi conexe sunt grupate într-o singură familie. Deci, s-ar putea spune că, de exemplu, familia de fonturi Verdana este fontul Verdana Bold. Familia de fonturi determină caracteristicile vizuale ale fontului. Astfel, toate fonturile dintr-o familie au caracteristici vizuale comune. Ceea ce le diferențiază se poate numi variație pe aceeași temă. Câteva dintre cele mai semnificative

variații sunt: light, italic, regular, bold etc.

Foarte des, termenul de familie de fonturi este echivalat cu termenul de font, și astfel se poate auzi: fontul Verdana. Acest lucru este complet legitim și, în cele mai multe cazuri, înseamnă o variație regulată sau normală dintr-o familie de fonturi.

## Proprietățile fonturilor pe web

Până acum, în acest program, singura caracteristică a fontului pe care am influențat-o în mod direct a fost dimensiunea caracterelor. Toate celelalte caracteristici ale fontului utilizat depindeau în întregime de browserul în care erau afișate documentele noastre HTML. Acest lucru înseamnă, practic, că pentru afișarea textului a fost folosit fontul implicit al browserului (în Google Chrome, fontul implicit este Times New Roman). Pentru a putea modifica proprietățile fontului utilizat în document, mai întâi trebuie să știți care sunt proprietățile care pot fi influențate (imaginea 17.2).



*Imaginea 17.2. Diferite proprietăți ale unui font*

În imagine sunt prezentate cele mai semnificative proprietăți de fonturi care pot fi influențate prin utilizarea CSS-ului. Acestea sunt proprietăți care afectează în mod direct aspectul unui caracter. Pe lângă aceste proprietăți, CSS mai are și alte proprietăți care pot fi influențate de cuvinte individuale, linii sau pasaje complete. Toate aceste proprietăți vor fi abordate în continuarea acestei lecții.

## Alegerea familiei de fonturi

Prima etapă în schimbarea proprietăților fontului unui document HTML implică definirea familiei sale. După cum am menționat, familia căreia îi aparține un font este o caracteristică de bază care determină caracteristicile sale vizuale. Din această cauză, termenul de familie de fonturi este adesea numit și font, așadar, în această lecție veți citi deseori termenul de font în loc de familie de fonturi.

Familia de fonturi poate fi setată folosind proprietatea `font-family`. Următorul exemplu ilustrează utilizarea acestei proprietăți:

```
body {  
  font-family: Georgia;  
}
```

Prin afișarea regulii CSS, familia de fonturi a documentului HTML complet este setată la `Georgia`.

În acest moment este foarte important să înțelegem că este necesar ca fontul definit să existe pe computerul utilizatorului. În caz contrar, nu va fi posibilă utilizarea acestuia și browserul va utiliza fontul implicit. Situația este aceeași ca, de exemplu, la documentele Word. Dacă nu există un font folosit pe computerul în care este vizualizat documentul Word, textul va fi afișat folosind un alt font disponibil pe sistem. Cu alte cuvinte, implicit, **fonturile nu sunt parte integrantă a unui site**

**sau a unei aplicații, ci sunt citite de pe computerul utilizatorului.** Cu alte cuvinte, întrebarea logică este; *cum să știu ce fonturi să folosesc atunci când creez site-uri și aplicații?* Într-adevăr, nu puteți fi niciodată sută la sută siguri că un font pe care îl definiți ca o proprietate de tip `font-family` va exista pe dispozitivele client. Cu toate acestea, există o serie de fonturi care sunt disponibile, cu siguranță, pe fiecare dispozitiv și, prin urmare, ele sunt numite fonturi web sigure (*web safe fonts*).

### Fonturile web sigure

O serie de fonturi sunt considerate fonturi web sigure, deoarece există pe aproape toate sistemele de operare utilizate în prezent (Windows, macOS, Linux, Android, iOS). Aceste fonturi sunt următoarele:

- Arial;
- Courier New;
- Georgia;
- Times New Roman;
- Trebuchet MS;
- Verdana.

Proprietatea CSS de tip `font-family` poate accepta multe valori. În această situație, se va utiliza valoarea primului font care există pe dispozitivul utilizatorului. Un exemplu de definire a unei proprietăți de tip `font-family` cu mai multe valori este următorul:

```
h1 {  
  font-family: "Times New Roman", Helvetica, sans-serif;  
}
```

Cu regula CSS prezentată, browserul va încerca să folosească fontul Times New Roman, iar dacă nu îl va găsi, va căuta fontul Helvetica. Dacă nici Times New Roman, nici Helvetica nu sunt disponibile, orice

font Sans Serif disponibil pe sistemul utilizatorului va fi utilizat, deoarece `sans-serif` este setat ca ultima valoare a proprietății `font-family`.

De asemenea, este foarte important să observăm o altă regulă importantă în exemplul prezentat. Dacă numele familiei de fonturi este format din mai mult de un cuvânt, acest nume este plasat în ghilimele.

## Fonturile generice

Exemplul tocmai prezentat a ilustrat pentru prima dată utilizarea uneia dintre valorile generice pe care `font-family` le poate accepta. Prin urmare, `sans-serif` nu este numele niciunui tip de font sau a unei familii de fonturi, dar este una dintre categoriile prezentate la începutul acestei lecții.

CSS permite setarea mai multor nume generice de fonturi pentru valoarea proprietății `font-family`, iar unele dintre cele mai importante sunt:

- Serif;
- Sans Serif;
- Monospace.

Folosind nume generice de font, alegerea unui anumit font depinde doar de browser. Singurul lucru pe care browserul trebuie să îl completeze la selectarea acestuia este ca fontul selectat să aparțină unei categorii definite de o valoare generică. Astfel, dacă specificați o valoare generică `sans-serif`, browserul își va selecta fontul implicit din grupul de fonturi Sans Serif.

Valorile generice sunt cel mai des specificate ca ultimele, așa cum este prezentat în exemplul precedent. În acest fel, acestea vor fi activate numai dacă browserul nu găsește niciunul dintre fonturile predefinite.



## Grosimea fontului - Bold

Următoarea caracteristică a fontului pe care o vom lua în considerare este legată de grosimea caracterelor. Ea este definită folosind proprietatea CSS `font-weight`. Această proprietate definește cât de subțiri sau groase vor fi caracterele fontului utilizat. Un exemplu de utilizare a acestei proprietăți este următorul:

```
p{  
font-weight: bold;  
}
```

Regula CSS dată indică faptul că textul din toate elementele de tip paragraph de pe pagină va fi scris cu caractere aldine/îngroșate (bold).

Proprietatea `font-weight` poate accepta patru valori text diferite și nouă valori numerice. Astfel, una dintre următoarele valori text poate fi utilizată pentru definirea grosimii caracterelor fontului:

- `normal`;
- `bold`;
- `bolder`;
- `lighter`.

Pe lângă valorile text, proprietatea `font-weight` poate accepta valori numerice de la 100 la 900, cu pasul 100. Valoarea 400 are același efect ca o valoare normală, în timp ce valoarea 700 este identică cu o valoare aldină/`bold`. Un exemplu de definire a grosimii caracterelor folosind o valoare numerică este următorul:

```
p{  
font-weight: 900;  
}
```

Într-una dintre lecțiile anterioare în care au fost abordate diferite elemente de marcare HTML, au fost prezentate atât elemente `b`, cât și elementele `strong`. Acestea sunt elementele care primesc în mod implicit următoarea structură:

```
strong, b {  
  font-weight: bold;  
}
```

Este recomandat să utilizați o regulă CSS pentru a seta text cu caractere *aldine*. Cu toate acestea, dacă o anumită parte a textului trebuie subliniată semantic, este bine să plasați un astfel de text într-un element `strong`.

## Italic

Pentru a formata un text ca italic, este posibil să folosiți proprietatea `font-style`. Următorul exemplu ilustrează utilizarea acestei proprietăți:

```
p {  
  font-style: italic;  
}
```



Proprietatea `font-style`, pe lângă valorile `italic`, poate avea valori normale și oblice (`normal` și `oblique`). Valoarea implicită a acestei proprietăți este normală, ceea ce înseamnă că formatarea inițială italică nu se aplică. Valoarea oblică (`oblique`) definește un efect `italic` mai intens prin crearea înclinării suplimentare.

## Dimensiunea textului

Alegerea dimensiunii textului poate fi realizată folosind proprietatea `font-size`. Următorul exemplu ilustrează definirea dimensiunii fontului:

```
body{  
  font-size: 12px;  
}
```

Regula CSS prezentată stabilește dimensiunea textului pentru conținutul text complet al documentului HTML. Dimensiunea textului este exprimată folosind valorile în pixeli. Pe lângă valorile exprimate în pixeli, CSS permite definirea dimensiunii textului folosind valori exprimate în diverse alte unități. Astfel, toate unitățile de exprimare a dimensiunii pot fi împărțite în unități absolute și relative.

Unitățile absolute sunt:

- in – inch (2,54cm);
- pt – point, punctul tipografic (1 point este 1/72 inch);
- pc – pica (1 pica este 12 puncte);
- cm – centimetru;

- mm – milimetru;
- px – [pixel](#).

Unitățile relative sunt:

- % – procentul;
- em-ul;
- rem-ul.

## Unitățile absolute și relative

Unitățile absolute sunt utilizate pentru determinarea absolută a dimensiunii textului. Cu alte cuvinte, dimensiunea absolută a literelor va fi întotdeauna aceeași, indiferent de dimensiunea afișării sau de orice alt parametru. Astfel de unități sunt, în general, preluate din presa scrisă, cu excepția pixelilor, care sunt caracteristici pentru afișarea digitală a imaginii.

Unitățile relative sunt cele care își formează valoarea finală pe baza dimensiunii principale a textului. De exemplu, dimensiunea implicită a textului în majoritatea browserelor este de 16 px. Astfel, se poate spune că 1 em = 16 px. Dacă dimensiunea implicită a textului este mărită la 20 px, 1 em va avea o valoare de 20 px. Similar este și în cazul procentelor, 100% reprezintă 16px. Pentru a înțelege corect diferența dintre unitățile absolute și cele relative, în imaginile 17.3 și 17.4 sunt ilustrate diferitele situații și efecte pe care le produc.

	<code>body {font-size:100%;}</code>
<code>font-size:1em</code>	lorem ipsum dolor sit amet
<code>font-size:16px</code>	lorem ipsum dolor sit amet
<code>font-size:100%</code>	lorem ipsum dolor sit amet

*Imaginea 17.3. Utilizarea diferitor unități pentru setarea dimensiunii fontului*

În imaginea 17.3 se poate observa că pe elementul body este setată dimensiunea textului la 100%, folosind unitățile relative. Într-o astfel de situație, dimensiunea textului descendenților direcți ai elementului body setat în moduri diferite, va fi ca în tabelul din imaginea 17.3. Se poate concluziona că dimensiunea textului este aceeași pentru toate cele trei valori (1em, 16 px, 100%).

Diferența reală între unitățile relative și absolute este vizibilă atunci când mărimea textului pe elementul body este mărită. Aceasta este exact situația ilustrată în imaginea 17.4.

	<code>body {font-size:130%;}</code>
<code>font-size:1em</code>	lorem ipsum dolor sit amet
<code>font-size:16px</code>	lorem ipsum dolor sit amet
<code>font-size:100%</code>	lorem ipsum dolor sit amet

*Imaginea 17.4. Utilizarea diferitor unități pentru setarea dimensiunii fontului (2)*

Dacă dimensiunea textului pe elementul body este mărită la 130%, se poate observa că textul cu unități absolut exprimate rămâne neschimbat, în timp ce textul cu dimensiuni relative urmărește în mod

adecvat redimensionarea pe elementul părinte de tip body.

## Em-ul și procentele

Când vine vorba de definirea dimensiunii unui text, se poate spune că procentele și unitățile em se comportă la fel, prin urmare, utilizarea unităților em va fi prezentată în continuare, iar tot ceea ce va fi prezentat se aplică și procentelor. Procentele și unitățile em sunt relative în raport cu elementul părinte direct. Acest lucru este ilustrat de exemplul prezentat în imaginea 17.5.



*Imaginea 17.5. Cum funcționează unitățile em?*

Structura elementelor prezentate în imagine este următoarea:

```
<body style="font-size:1em;">  
  body text  
  <div style="font-size:3em;">  
    first div text  
    <div style="font-size:0.5em;">  
      second div text  
    </div>  
  </div>  
</body>
```

În interiorul părții `body` se află un `div` și în interiorul său un alt `div`. Toate aceste elemente au conținut textual. Elementul `body` are dimensiunea textului de  $1\text{ em}$ . Primul descendent direct al elementului `body` este setat la un text de  $3\text{ em}$ . Pe `div`-ul imbricat este setată dimensiunea textului la  $0,5\text{ em}$ . Din imaginea 17.5 se poate concluziona că dimensiunea textului din ultimul `div` este mai mare decât textul găsit direct în interiorul elementului `body`, deoarece dimensiunea acestuia este relativă cu elementul părinte. Dimensiunea textului ultimului element `div` din structura prezentată poate fi calculată după cum urmează (dacă dimensiunea implicită a textului este de 16 px):

$$16\text{px} * 3 * 0.5 = 24\text{px}$$

Din această calculație se poate trage concluzia că în calculul dimensiunii fontului din elementul `div` imbricat, au fost implicate dimensiunile literelor tuturor elementelor părinte.

## Unitatea `rem`

O altă unitate relativă care poate fi utilizată pentru definirea dimensiunii textului în documentele HTML este `rem`. Este vorba, de fapt, de o unitate interpretată ca *Root EM*, de unde și provine numele ei - `rem`.

Unitatea `rem` este foarte asemănătoare cu unitatea `em` deja descrisă. Diferența constă doar în elementul de bază care este luat ca referință și pe baza căruia va fi calculată valoarea finală. La unitatea `em`, acesta este un element părinte direct. Pe de altă parte, unitatea `rem` este întotdeauna relativă cu elementul rădăcină al documentului HTML - `<html>`. Astfel, atunci când se utilizează unitatea `rem`, calculul final nu include valorile găsite pe toate elementele părinte din ierarhie, ca în

imaginea 17.5. Singurul element care este considerat relevant este elementul `html`. Desigur, cu excepția cazului în care elementul `html` definește dimensiunea textului, dimensiunea de bază a browserului este luată ca bază, care este în general de 16 px. Caracteristicile unităților `rem` sunt ilustrate în imaginea 17.6.



*Imaginea 17.6. Cum funcționează unitățile `rem`?*

Un exemplu identic cu cel precedent a fost utilizat pentru a demonstra proprietățile unităților `rem`. Documentul HTML are o dimensiune de text de bază de 16 px. Aceasta este și dimensiunea textului în interiorul elementului `body`. Dimensiunea textului primului element `div` nu se schimbă dacă sunt utilizate unitățile `em` sau `rem`. Cu toate acestea, există diferențe cu privire la ultimul element `div` din structură. Când utilizați unități `rem`, dimensiunea textului său este 8 px, deoarece este relativă cu dimensiunea textului elementului `html`, care este de 16 px.

## Notă

Toate unitățile de mai sus, care pot fi utilizate pentru definirea dimensiunii fontului, pot fi, de asemenea, utilizate pentru definirea valorilor altor proprietăți, cum ar fi marginile, înălțimea, lățimea și altele. Prin urmare, este foarte important să înțelegem diferența dintre unitățile relative și cele absolute, deoarece acestea vor fi utilizate intensiv în continuarea cursului.

## Culoarea textului

Culoarea textului dintr-un document HTML poate fi definită folosind proprietatea `color` a regulii CSS. Valoarea poate fi exprimată folosind toate abordările descrise într-una din lecțiile anterioare ale acestui modul.

HEX:

```
h1 {  
color: #00FF00;  
}
```

Denumirile date culorilor:

```
h1 {  
color: green;  
}
```

RGB:

```
h1 {  
color: rgb(0,255,0);  
}
```



HSL:

```
h1 {  
  color: hsl(120, 100%, 50%);  
}
```

Toate exemplele prezentate ilustrează setarea culorii textului `h1` la verde.

## Alinierea textului

Proprietatea `text-align` este utilizată pentru alinierea textului orizontal. Valorile disponibile pentru alinierea orizontală sunt: `left`, `right`, `center` și `justify`. Valorile stânga și dreapta (`left` și `right`) vor alinia textul de-a lungul marginilor din stânga și dreapta cadrului. Valoarea `center` va centra textul, în timp ce valoarea `justify` va alinia textul atât pe marginea stângă, cât și pe cea dreaptă, influențând astfel spațiile dintre cuvinte.

Următorul exemplu ilustrează utilizarea proprietății de aliniere a textului:

```
#footer {  
  text-align: center;  
}
```

Pe lângă alinierea orizontală, CSS permite alinierea textului de-a lungul axei **verticale**. Alinierea pe axa verticală se realizează folosind

proprietatea `vertical-align`. Această proprietate permite, de fapt, alinierea verticală a textului diferit într-o singură linie. Un exemplu de utilizare a acestei proprietăți este și următorul:

```
.textBaseline {  
  vertical-align: baseline;  
}
```

Exemplul ilustrează alinierea textului de-a lungul liniei de bază a unui element de tip container. Inclusiv valoarea de bază/baseline specificată, pentru valoarea proprietății `vertical-align` se pot specifica următoarele valori:

- `baseline` - aliniază textul de-a lungul liniei părinte;
- `top` - aliniază partea superioară a textului cu partea superioară a elementului cel mai înalt din linie;
- `text-top` - aliniază partea superioară a textului cu partea superioară a elementului părinte;
- `middle` - plasează textul în mijlocul elementului părinte în înălțime;
- `bottom` - aliniază partea inferioară a textului cu partea inferioară a elementului cel mai de jos din linie;
- `text-bottom` - aliniază partea inferioară a textului cu partea inferioară a elementului părinte.

## Indentarea textului

Indentarea textului poate fi obținută folosind proprietatea `padding-left`

`text-indent`. Această proprietate poate accepta câteva dintre valorile absolute descrise anterior, unde valoarea reflectă cât de bine va fi indentat textul. Un exemplu de utilizare a proprietății de indentare a textului este următorul:

```
p {  
  text-indent: 50px;  
}
```

Valoarea proprietății `text-indent` poate fi, de asemenea, exprimată în procente și în acest caz ea se referă la lățimea containerului părinte în care se află textul. Acest lucru este ilustrat în următorul exemplu:

```
p {  
  text-indent: 10%;  
}
```

## Spațiul dintre cuvinte și caractere

Prin utilizarea CSS-ului se poate controla foarte ușor spațiul dintre cuvintele și caracterele din text. Proprietatea `letter-spacing` se utilizează pentru controlul spațiului dintre caractere, în timp ce proprietatea `word-spacing` este utilizată pentru definirea spațiului dintre cuvinte. Următorul exemplu ilustrează utilizarea acestor două proprietăți:

```
p {  
  letter-spacing: 8px;  
  word-spacing: 16px;  
}
```

## Spațiul dintre rândurile textului

Pe lângă spațiul dintre caractere și cuvintele din text individuale, utilizarea CSS-ului poate influența și spațiul dintre liniile de text. Mai precis, CSS vă permite să definiți înălțimea unei linii de text, care influențează în mod direct spațiul dintre linii. Proprietatea `line-height` este utilizată pentru definirea înălțimii unei linii de text, iar utilizarea acestei proprietăți este prezentată prin următorul exemplu:

```
p{  
  line-height:1.4em;  
}
```

## Transformarea textului

Când vine vorba de transformarea textului, CSS-ul se referă în primul rând la utilizarea proprietății `text-transform`. Această proprietate poate avea una dintre următoarele valori:

- `uppercase`;
- `lowercase`;
- `capitalize`.

Proprietățile prezentate transformă textul definit în element prin transformarea tuturor literelor în minuscule sau majuscule sau prin plasarea primei litere majuscule a fiecărui cuvânt. Este destul de clar că toate literele sunt transformate în majuscule folosind valoarea `uppercase` și că toate literele sunt transformate în minuscule folosind

proprietatea `lowercase`. Prima literă a fiecărui cuvânt din text se obține folosind valoarea `capitalize`.

Următorul exemplu ilustrează utilizarea proprietății `text-transform`:

```
h1 {  
  text-transform: uppercase;  
}
```

```
<h1>this is Heading</h1>
```

Exemplul prezentat definește proprietatea `text-transform` cu valoarea `uppercase`, care va fi aplicată tuturor titlurilor `h1`. Un astfel de titlu este dat și în exemplu. La randarea unei pagini, titlul va fi afișat în felul următor:

THIS IS HEADING

Se poate observa că toate literele titlului au fost convertite în majuscule. Este important să înțelegeți că în documentul HTML sursă, textul rămâne așa cum este și că se transformă doar vizualizarea pentru utilizator. De aceea această proprietate se și numește `text-transform`.

## Decorarea textului

Decorarea textului presupune plasarea unei linii care poate fi găsită mai jos, deasupra sau în mijlocul textului. Pentru a realiza acest lucru,

se utilizează proprietatea `text-decoration`, care poate avea următoarele valori:

- `none;`
- `underline;`
- `overline;`
- `line-through.`

Numele valorilor sunt destul de ilustrative, iar exemplul care ilustrează utilizarea acestora este următorul:

```
a
{
  text-decoration: none;
}
```

În exemplul de mai sus este stilizată afișarea textului unui link. În exemplu nu am luat întâmplător tocmai stilizarea unui link, având în vedere faptul că în mod implicit, linkurile de pe pagină se afișează subliniate. Dacă trebuie eliminată o astfel de caracteristică, se poate folosi regula CSS tocmai prezentată.

## Umbra textului

Folosind CSS este posibil să setați și umbra pe care textul o plasează pe elementul de mai jos. Acest lucru este realizat folosind proprietatea `text-shadow`. Un exemplu de utilizare a acestei proprietăți este următorul:

```
h1 {
  text-shadow: 2px 2px #ff0000;
}
```

În exemplu, se poate observa că valoarea proprietății `text-shadow` este formată din mai multe părți. Sintaxa acestei proprietăți este următoarea:

`text-shadow: h-shadow v-shadow blur-radius color;`

Din analiza sintaxei se poate concluziona că prima parte a valorii se referă la poziția orizontală a umbrei, a doua parte se referă la poziția sa verticală, a treia parte la diametrul efectului de estompare/blur, în timp ce ultima parte a valorii se referă la culoarea umbrei.

## Utilizarea fonturilor inexistente pe computerul utilizatorului

Toate exemplele de până acum ilustrau utilizarea fonturilor găsite pe computerul utilizatorului. Totuși, în documentele HTML puteți să utilizați și fonturi care nu se află pe dispozitivul utilizatorului. Această abordare este foarte utilă dacă trebuie să utilizați fonturi non-standard, care nu aparțin listei de fonturi web sigure, în documentul HTML.

Cele mai frecvente două abordări pentru utilizarea fonturilor care nu se află pe sistemul client sunt:

- găzduirea independentă a fontului prin plasarea fizică a fișierelor font la aceeași locație pe care se află site-ul sau aplicația;
- utilizarea unora dintre fonturile web.



## Găzduirea independentă a fontului

Găzduirea independentă a unui font implică plasarea fișierelor fontului la aceeași locație la care se află site-ul sau aplicația. Pentru a utiliza un astfel de font, locația fontului trebuie definită folosind un cod CSS special:

```
@font-face {  
font-family: "My Nicer Font";  
src: url('fonts/my_woff_font.eot');  
src: url('fonts/my_woff_font.woff');  
}
```

Construcția specială `@font-face` este, de fapt, un bloc de cod, în cadrul căruia sunt setate valori ale diferitor proprietăți. Proprietatea `font-family` definește numele fontului, adică familiile de font. Proprietatea `src` definește locația fontului. De asemenea, fonturile pot fi definite prin fișiere cu diferite extensii. În exemplu sunt prezentate fișierele cu extensiile `eot` și `woff` și, în plus, există fișiere care pot avea extensiile `ttf`, `otf` sau `svg`. Trebuie să știm că diferite browsere acceptă fonturi de diferite formate (tabelul 17.1), deci este mai bine să avem același font în mai multe formate, la fel cum este cazul și în exemplu (EOT pentru versiunile mai vechi de Internet Explorer și WOFF pentru celelalte).

Format	Suport
EOT (Embedded OpenType)	IE
WOFF (Web Open Font Format)	IE, Edge, Firefox, Chrome, Safari, Opera, iOS, Android
SVG	Safari, iOS

TTF (TrueType), OTF (OpenType)	Edge, Firefox, Chrome, Safari, Opera, iOS, Android
WOFF 2.0 (Web Open Font Format)	Edge, Firefox, Chrome, Safari, Opera, iOS, Android

*Tabelul 17.1. Diferite formate de fonturi care pot fi înțelese de browsere*

Fontul definit prin folosirea construcției `@font-face` poate fi utilizat într-unul dintre elementele HTML, în același mod ca și fonturile care există pe dispozitivul utilizatorului:

```
p {  
  font-family: 'My Nicer Font';  
}
```

## Utilizarea fonturilor web

Un alt mod de utilizare a fonturilor care nu se folosesc pe dispozitivele client este să folosiți așa-numitele fonturi web. Aceasta este o abordare care este foarte populară pe web-ul modern de astăzi și chiar Google are un spațiu de stocare cu multe astfel de fonturi care pot fi utilizate gratuit (<https://fonts.google.com/>). Utilizarea lor se bazează pe includerea unui anumit font în documentul HTML folosind elementul link:

```
<link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">
```

Linia prezentată ilustrează includerea fontului Roboto într-un document HTML. Includerea se face folosind elementul `link`, care trebuie plasat în antetul documentului (secțiunea head).

Fontul web inclus în acest mod poate fi folosit pe oricare dintre elementele HTML, la fel ca și orice alt font de până acum:

```
p {  
  font-family: 'Roboto';  
}
```

## HTCF9\_17 - HTML & CSS Fundamentals

1. În imagine este prezentat fontul care intră în categoria:

HTML & CSS

- ☐ a) serif
- ☐ b) monospaced
- ☐ c) sans serif
- ☐ d) cursive

2. Cu ajutorul proprietății font-weight se poate seta:

- ☐ a) un text overline (peste care este trasă o linie)
- ☐ b) un text underline (subliniat)
- ☐ c) un text bold (alidin/îngroșat)
- ☐ d) un text italic (înclinat)

3. Proprietatea text-indent se folosește pentru:

- ☐ a) indentarea primului rând de text pe orizontală
- ☐ b) determinarea dimensiunii textului
- ☐ c) determinarea înălțimii rândurilor în text
- ☐ d) alinierea stânga/dreapta/centrală a textului

4. Dimensiunea implicită a textului în majoritatea browserelor este de:

- ☐ a) 16 px
- ☐ b) 14 px
- ☐ c) 12 px
- ☐ d) 18 px

5. Dacă pentru dimensiunea textului unui element este setată valoarea de 2.5 em, iar partea body are dimensiunea textului de 12 px, dimensiunea textului acestui element care se găsește direct în cadrul părții body a paginii va fi de:

- ☐ a) 30 px
- ☐ b) 20 px
- ☐ c) 26 px
- ☐ d) 12 px

6. Completați următorul text cu noțiunile care lipsesc:

Răspunsuri oferite: fonturile sans serif, fonturile monospace, fonturile serife

\_\_\_\_\_ sunt fonturi caracterizate prin lipsa detaliilor pe care le dețin

## 7. Câteva dintre unitățile relative sunt:

- ☐ a) px – pixelul
- ☐ b) mm – milimetrul
- ☐ c) em – em-ul
- ☐ d) % – procentul
- ☐ e) pt – point-ul (punctul)

**1. În imagine este prezentat fontul care intră în categoria:**

a

**2. Cu ajutorul proprietății font-weight se poate seta:**

c

**3. Proprietatea text-indent se folosește pentru:**

a

**4. Dimensiunea implicită a textului în majoritatea browserelor este de:**

a

**5. Dacă pentru dimensiunea textului unui element este setată valoarea de 2.5 em, iar partea body are dimensiunea textului de 12 px, dimensiunea textului acestui element care se găsește direct în cadrul părții body a paginii va fi de:**

a

**6. Completați următorul text cu noțiunile care lipsesc:**

fonturile sans serif, fonturile serife

**7. Câteva dintre unitățile relative sunt:**

c, d