

**TEHNICI WEB**

**CSS #3**

*Claudia Chiriță . 2024/2025*

# CSS LAYOUT: MULTICOLOANE

- permite aranjarea conținutului pe mai multe coloane, ca într-un ziar
- în esență, un container cu proprietatea *column-count* sau *column-width*

```
column-count: numărul de coloane
```

```
column-width: lățimea unei coloane;
```

# MULTICOL

- proprietăți pentru controlarea stilului într-un layout multicol:

```
column-gap: spațiul dintre coloane
```

```
column-span: câte coloane ocupă un element;
```

```
column-rule: stilul liniei care desparte coloanele;  
/* prescurtare column-rule: column-rule-style  
        column-rule-width column-rule-color*/
```

# CSS ACROSS DEVICES

- adaptarea layoutului unei pagini web la diversele dispozitive ale utilizatorilor:  
mobile (telefon, tabletă) & calculator

Adaptive/Mobile Friendly Design

VS.

Responsive Web Design

# ADAPTIVE DESIGN

- aplicația server detectează dispozitivul utilizatorului și răspunde cu pagini diferite
- ex: webpage.com vs. m.webpage.com

# RESPONSIVE DESIGN

o singură versiune a paginii care funcționează pe orice dispozitiv

avantaje:

- mai ușor de menținut
- SEO friendly
- posibilitatea de sharing a aceluiași URL între diverse dispozitive

# MEDIA QUERIES

- aplicarea unui stil CSS în funcție de tipul general al unui dispozitiv sau alte caracteristici precum rezoluția ecranului sau lățimea browserului
- aplicarea condiționată a unui stil folosind regulile @media și @import
- pentru specificarea media pentru elemente precum <style>, <link>, <source> folosind atributul media=
- pentru accesibilitate

# MEDIA QUERIES

```
media type: screen, print, speech
```

```
media features: width, height, max-height, max-width,  
                orientation, resolution, aspect-ratio
```

sintaxă media query:

```
media type and (med-feature1:val) and (med-feature2:val)...
```



# MEDIA QUERIES HTML

- atributul HTML *media*
- folosirea unor stiluri CSS diferite în funcție de proprietățile dispozitivului media

```
<head>  
  <link rel="stylesheet" type="text/css" media="screen and (ma  
  <link rel="stylesheet" type="text/css" media="print" href="p  
</head>
```

# MEDIA QUERIES CSS

- CSS at-rule @media

```
@media media-query1, media-query2
```

exemplu:

```
@media screen and (max-width: 960px) and (min-width: 500px)
{
  body { background-color: pink; }
}
```

## OPERATORI @MEDIA

and, only, not și ","

- *only* e folosit pentru browsere mai vechi care nu-l recunosc, css-ul din media query fiind ignorat  
@media only x este echivalent cu @media x
- operatorul "," are rol de disjuncție

# VIEWPORT

```
<head>  
  <meta name="viewport" content="width=device-width, initial  
</head>
```

indică browserului că pagina trebuie să fie scalată  
pentru a se potrivi cu dimensiunea ecranului

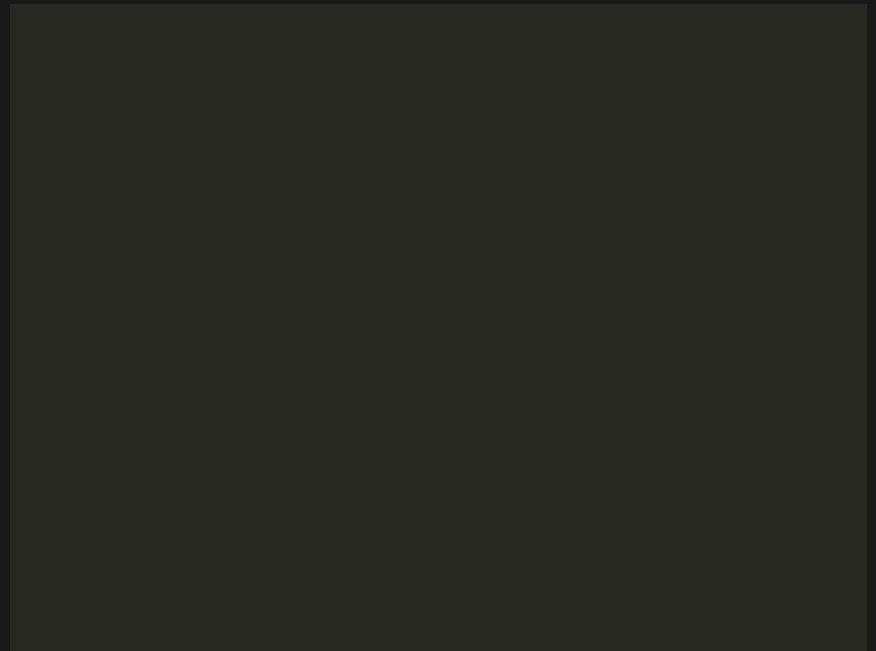
# UNITĂȚI DE MĂSURĂ RELATIVE

- unități de lungime relative la dimensiunea elementului părinte / a fontului elementului părinte, ori a dimensiunii viewportului
- ex: em, rem, vw, vh, vmin, vmax

# TEXT RESPONSIVE

- folosind unități de măsură ale viewportului (vw, vh, vmin, vmax)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      h1 {font-size: 5vw;}
      h2 {font-size: 4vw;}
      p {font-size: calc(0.5rem + 1vw);}
    </style>
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <p>Paragraph</p>
  </body>
</html>
```



# TEXT RESPONSIVE

- folosind media queries

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      @media screen and (min-width:768px){
body{font-size:20px;}
      }
      @media screen and (max-width:768px){
body{font-size:12px;}
      }
      h1 {font-size: 3em;}
      h2 {font-size: 1.5em;}
      p {font-size: 0.9em;}
    </style>
  </head>
```



# IMAGINI RESPONSIVE

- elementele `img`, `video`, `iframe` nu se redimensionează cu cod HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #container{
        width:80%;
        border: 2px solid red;
        float: left;}
    </style>
  </head>
  <body>
    <div id="container">
      
      <p>The red panda (Ailuropus
```





# IMAGINI RESPONSIVE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #container{
        width:80%;
        border: 2px solid red;
        float: left;}
      img{
        max-width:100%;
        height:auto;}
    </style>
  </head>
  <body>
    <div id="container">
      
```



# IMAGINI RESPONSIVE

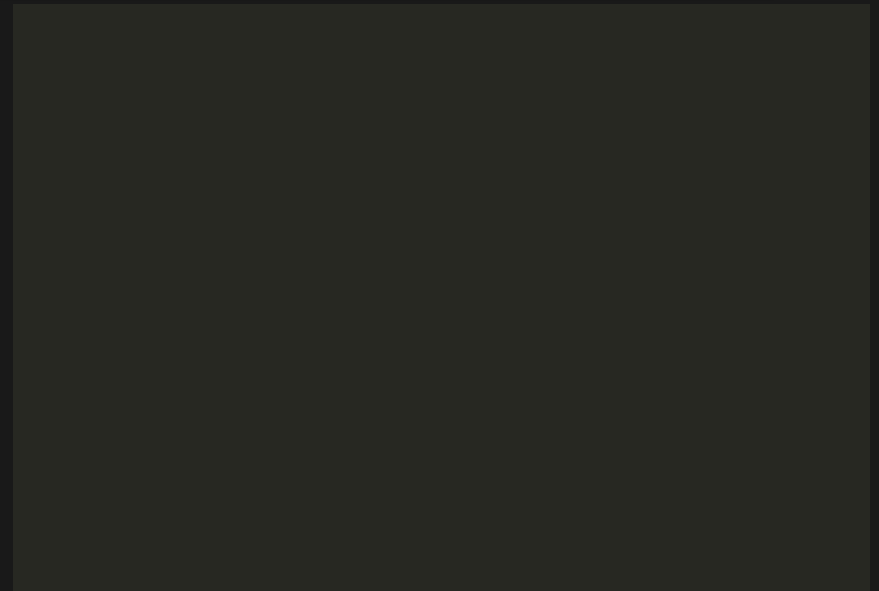
pentru dimensiuni diferite se vor afișa imagini diferite;  
folosind elementele html <picture> și <source>

```
<picture>  
  <source srcset="small.jpg" media="(max-width:500px)">  
  <source srcset="med.jpg" media="(max-width:1000px)">  
    
</picture>
```

# IMAGINI BACKGROUND RESPONSIVE

```
background-image: url(adresă-imagine);  
background-size: cover;  
/* imaginea acoperă întregul container*/
```

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <link rel="stylesheet"  
href="demo.css">  
    <style type="text/css">  
      body {margin-left:auto;  
margin-right:auto; width:90%;  
background-image:url(boo.jpg);  
background-repeat: no-repeat;  
background-size:cover;}  
    </style>  
  </head>  
  <body>  
  </body>
```



# FLASH QUIZ: ALINIERI

centrare pe orizontală

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square { background-color:
khaki;
        color: black;
        height: 150px; width: 150px;
        /* margin-left: ?;
        margin-right: ?; */
      }
      a { /*margin: ?;
        width: 100?;
        display: ?*/
    }
```



# FLASH QUIZ: ALINIERI

to the right!

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square { background-color:
khaki;
        color: black;
        height: 150px; width: 150px;
        /*position: ?; */
      }
    </style>
  </head>
  <body>
    <div id="square">The Square</div>
  </body>
```



# FLASH QUIZ: ALINIERI

centrare pe verticală

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square { background-color:
khaki;
        color: black;
        height: 150px; width: 150px;
        /* display: ?;
        vertical-align: ?;*/
      }
      #parinte{ line-height:300px;
        height: 300px; }
    </style>
  </head>
```



**TRANSFORMĂRI**

# TRANSFORMĂRI CSS

- schimbarea poziției și forme conținutului unui element HTML fără afectarea fluxului normal al documentului
- transformări afine ale elementelor (box model)
- rotații, scalări, translații, forfecări în plan și 3D



# TRANSFORM

```
transform: metodă(valori);
```

- metode de transformare 2D:

```
translate(), rotate(), scale(), skew(), matrix()
```

- listă de transformări separate prin spațiu;  
transformările sunt aplicate pe rând

- ```
transform-origin
```

specifică poziția originii

# TRANSFORM: TRANSLATE

```
transform: translate(tx,ty)
          translateX(tx), translateY(ty);
```

mută elementul de la poziția inițială, conform parametrilor dați: [axa-X] și [axa-Y]

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square { background-color:
khaki;
      color: black;
      height: 150px; width: 150px;
      /*transform:
translate(10px,30px);*/
      }
    </style>
  </head>
  <body>
    <div id="square">The Square</div>
```

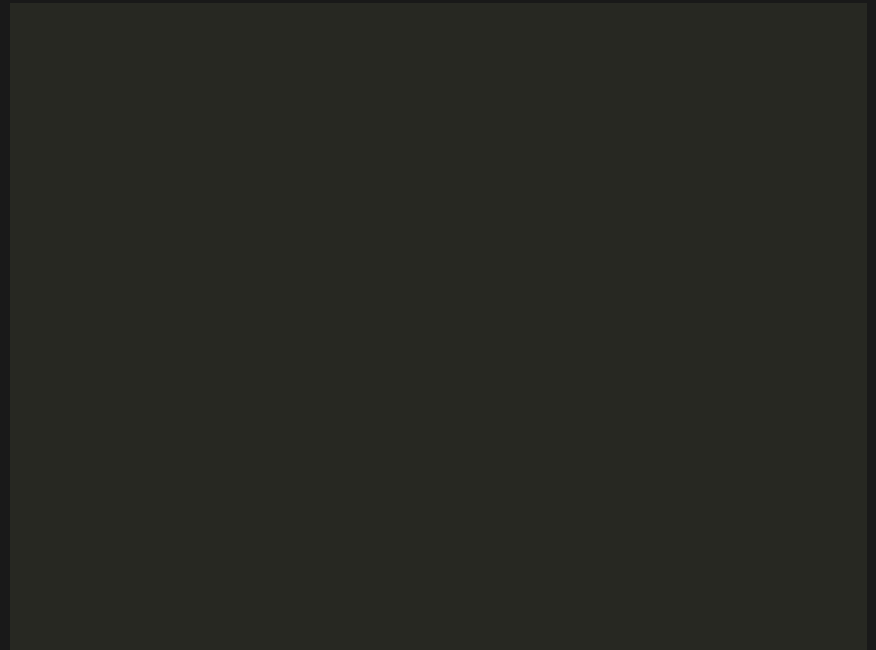


# TRANSFORM: ROTATE

```
transform: rotate(grade);
```

rotește elementul în sensul acelor de ceasornic dacă valoarea grade este pozitivă și invers dacă valoarea este negativă

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square { background-color:
khaki;
      color: black;
      height: 150px;
      width: 150px;
      /*rotate: -45deg;*/
    }
  </style>
</head>
<body>
  <div id="square">The Square</div>
```

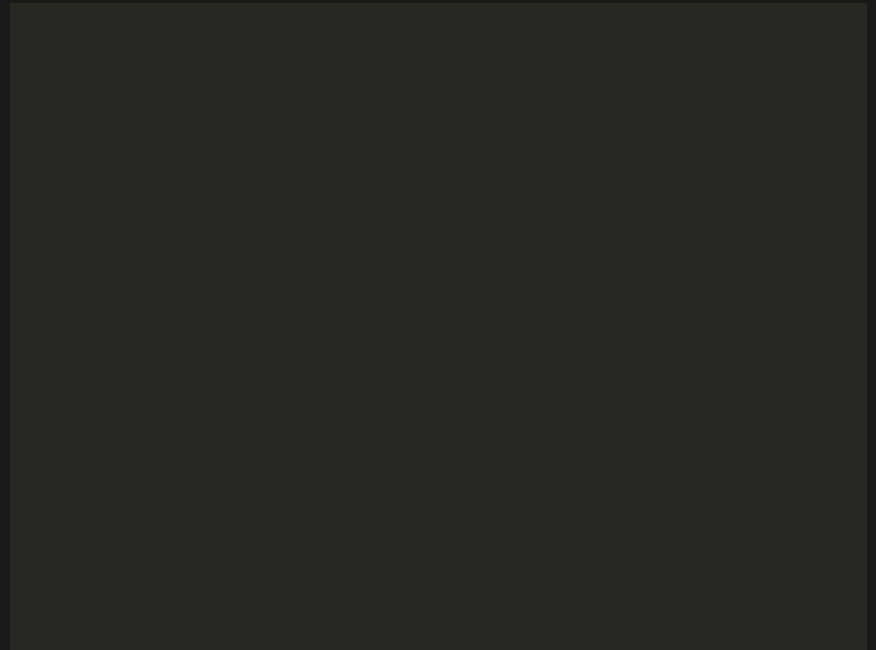


# TRANSFORM: SCALE

```
transform: scale(sx,sy),  
           scaleX(sx), scaleY(sy);
```

mărește sau micșorează dimensiunile elementului în funcție de parametrii *sx* și *sy*

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet"  
href="demo.css">  
    <style type="text/css">  
      #square { background-color:  
khaki;  
        color: black;  
        height: 150px; width: 150px;  
        /*transform: scale(0.66);*/  
      }  
    </style>  
  </head>  
  <body>  
    <div id="square">The Square</div>  
  </body>
```

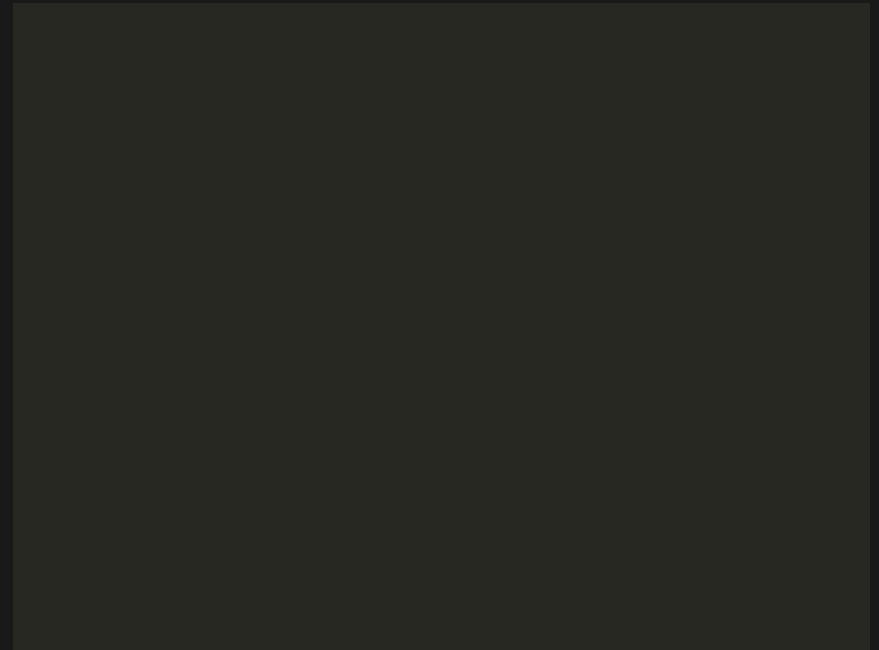


# TRANSFORM: SKEW

```
transform: skew(ax, ay),  
          skewX(ax), skewY(ay);
```

distorsionează elementul de-a lungul axelor X și Y în funcție de parametrii specificați

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet"  
href="demo.css">  
    <style type="text/css">  
      #square { background-color:  
khaki;  
        color: black;  
        height: 150px; width: 150px;  
        /*transform: skew(12deg,  
12deg);*/  
      }  
    </style>  
  </head>  
  <body>  
    <div id="square">The Square</div>
```



# TRANSFORM

unui element îi pot fi aplicate mai multe transformări  
în același timp

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square { background-color:
khaki;
      color: black;
      height: 150px; width: 150px;
      /*transform: translate(25px,30px)
rotate(45deg) scale(1.5);*/
      }
    </style>
  </head>
  <body>
    <div id="square">The Square</div>
```



# ACCESIBILITATE

- animațiile cu scalare/zoom pot ridica probleme din punct de vedere al accesibilității, putând provoca migrene
- e recomandată adăugarea unui mijloc de control pentru oprirea animațiilor dintr-un site
- e recomandată folosirea feature-ului media `prefers-reduced-motion`

# TRANSFORMĂRI 3D

- extensii ale transformărilor 2D
- efectul 3D se obține folosind proprietatea *perspective* (distanța pe axa Z)

```
translateZ(tz) -> translate3d(tx,ty,tz)
```

```
rotateZ(az) -> rotate3d(x,y,z,grade)
```

```
scaleZ(sz) -> scale3d(sx,sy,sz)
```

```
matrix3d(n,n,...,n)
```

```
perspective(n)
```



# TRANZIȚII

- controlul vitezei de animație la schimbarea valorii unei proprietăți CSS: schimbarea poate să aibă loc de-a lungul unei perioade de timp, nu doar instantaneu
- declanșate de evenimente

```
:hover, :focus, :active
```

# TRANZIȚII

proprietățile necesare definirii unei tranziții:

```
transition-property: proprietatea CSS de schimbat;  
transition-duration: s; /* durata tranziției */  
transition-timing-function: linear; /* ease, ease-in,  
                                ease-out, step-start, step-end  
                                (funcții predefinite) */  
transition-delay: s; /* durata de așteptare înainte  
                    de începerea tranziției */  
  
transition: property duration timing-function delay;
```

proprietăți cărora li se poate aplica o tranziție

# TRANZIȚII

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square {background-color: khaki;
color: black;
height: 150px; width: 150px;
transition: width 2s, height 2s,
background-color 2s, transform
2s;}
      #square:hover { background-color:
pink;
height: 100px;
width: 100px;
transform: rotate(45deg); }
```



# ANIMAȚII CSS

- constau din două componente:
  - un stil care descrie animația
  - o mulțime de *keyframes* care indică stările inițiale și finale ale animației și posibile puncte intermediare

# KEYFRAMES

```
@keyframes nume-animatie {  
  0% {...}  
  25% {...}  
  ...  
  75% {...}  
  100% {...}  
}
```

```
@keyframes nume-animatie {  
  from {...} /* echivalent cu 0% */  
  to {...} /* echivalent cu 100% */  
}
```

# ANIMATION

```
animation-name: nume-animatie;
```

```
animation-duration: s; /* obligatorie */
```

```
animation-timing-function: linear /*ease, ease-in, ease-out*/
```

```
animation-delay: s;
```

```
animation-iteration-count: val; /* infinite */
```

```
animation-direction:normal /* alternate, reverse */
```

```
animation-fill-mode: none /* forwards, backwards, both */
```

# ANIMATION

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      #square {background-color: khaki;
        color: black;
        height: 150px; width: 150px;
        animation-name: thesquare;
        animation-duration: 6s;
        animation-iteration-count:
infinite;}

      @keyframes thesquare {
        0% {background-color: khaki;}
        25% {background-color:
yellow;}
        50% {background-color:
```



# THE BRB

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style>
      .brb {
        height: 350px;
        width: 350px;
        display: inline-block;
        margin: 60px 0 60px 0;
      }
      div { text-align:center; }
    </style>
  </head>
  <body>
    <div>
      <span class="brb"></span>
    </div>
  </body>
</html>
```





# THE BRB

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style>
      .brb {
        height: 350px;
        width: 350px;
        border-radius: 50%;
        display: inline-block;
        margin: 60px 0 60px 0;
      }
      div { text-align:center; }
    </style>
  </head>
  <body>
    <div>
      <span class="brb"></span>
    </div>
  </body>
</html>
```



# THE BRB

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style>
      .brb {
        height: 350px;
        width: 350px;
        border-radius: 50%;
        display: inline-block;
        margin: 60px 0 60px 0;
      }
      .brb:active {
        background-color: red;
      }
      div { text-align:center; }
    </style>
  </head>
  <body>
    <div>
      <span class="brb"></span>
    </div>
  </body>
</html>
```



# THE BRB

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style>
      .brb {
        height: 350px;
        width: 350px;
        border-radius: 50%;
        display: inline-block;
        margin: 60px 0 60px 0;
        background-color: red;
      }
      .brb:active {
        animation-name: thebrb;
        animation-duration: .5s;
      }

      @keyframes thebrb {
        0% {height: 350px; width:
350px;}
        50% {height: 330px; width:
330px; opacity:0.75;}
        100% {height: 350px; width:
350px; opacity:1;}
      }
      div { text-align:center; }
```



# THE BRB

```
<!DOCTYPE html>
<html lang="en">
  <link rel="stylesheet"
href="demo.css">
  <style>
    .brb {
      height: 350px;
      width: 350px;
      border-radius: 50%;
      display: inline-block;
      margin: 60px 0 60px 0;
      background-color: red;
    }
    .brb:active {
      animation-name: thebrb;
      animation-duration: 1.25s;
      animation-iteration-count:
infinite;
    }

    @keyframes thebrb {
      0% {height: 350px; width:
350px;}
      20% {height: 360px; width:
360px; opacity:0.75;}
      40% {height: 350px; width:
350px; opacity:1;}
      60% {height: 370px; width:
370px; opacity:0.75;}
      80% {height: 350px; width:
350px; opacity:1;}
      100% {height: 350px; width:
350px; opacity:1;}
    }
  </style>
</html>
```



• •  
(>\----/<)  
' '  
' '•  
/ q p \  
( >(\_Y\_)< )  
>'-'-'<-.  
/ \_ .== ,=. , - \  
/ , ) ' ( )  
; '••' --<  
: \ | )  
) ; \_ /  
• \_ \_ / \_ \_ . ' - \ \ \ \  
- - \ \ \ \

întrebări?

