

## Megoldás

1., 2., 3. Készítse el a feladat által kért képet, a képszerkesztés fejezetben található információk segítségével.

4. Készítsen egy másolatot a megadott „programozasi\_nyelvek\_forras.html”-ről, és nevezze el „programozasi\_nyelvek.html”-nek.

Ugyanúgy mint az előző feladatnál, ügyeljen rá, hogy az összes kép, illetve a .html fájlok ugyan abban a könyvtárban legyenek.

Végül nyissa meg egy szövegszerkesztőben az új „programozasi\_nyelvek.html”-t:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title></title>
</head>

<body>
  <p>Programozási nyelvek</p>
  ....
  <p>Az olvashatóság számít.</p>
</body>

</html>
```

5. a) Hasonlóan az előző feladatsorhoz, ezt érdemes a **<body>** elem változtatásával megtenni:

```
<head>
  <meta charset="utf-8">
  <title></title>
  <style>
    body {
      color: black;
      background-color: rgb(0, 1, 44);
    }
  </style>
</head>
```

b) Állítsuk be megfelelően css-el az összes link (<a> elem) állapotát:

```
<style>
  body {
    color: black;
    background-color: rgb(0, 1, 44);
  }

  a {
    color: #080146;
  }

  a:visited {
    color: #080146;
  }

  a:active {
    color: #080146;
  }

  a:hover {
    color: #080146;
  }
</style>
```

c) Állítsuk be a <title> elemet:

```
<head>
  <meta charset="utf-8">
  <title>Programozási nyelvek</title>
```

d) Ezt a <body> elem „font-family” attribútumával tudjuk a legegyszerűbben megtenni:

```
body {
  color: black;
  background-color: rgb(0, 1, 44);
  font-family: Arial, Helvetica, sans-serif;
}
```

6. A feladatot meg lehetne oldani a <body> elemmel, ha a feladat csak a középre igazítást, és a szélesség megváltoztatását kérné, viszont külön kell most tudnunk állítani a lap háttérének, és a középső sávnak a színét. Meg lehetne ezt oldani úgy is, hogy minden elemél egyesével állítjuk a színeket, viszont ez nem kívánatos, mivel sok hibalehetőséget hordoz magával, úgyhogy ehelyett érdemes a <body> elemen belül egy új elemet felvenni, ami magába foglalja az összes egyéb elemet, és ezt az elemet állítjuk be megfelelően.

Ilyen feladatra általában a <div> elemet szokták használni, és mindenképpen állítsunk be „class”, vagy „id” attribútumot:

```

<body>
  <div class="content">
    <p>Programozási nyelvek</p>
    ....
    <p>Az olvashatóság számít.</p>
  </div>
</body>

```

Majd készítsük el a css-t hozzá:

```

.content {
  background-color: rgb(255, 255, 255);
  width: 900px;
  margin-left: auto;
  margin-right: auto;

  border-left: 30px dashed rgb(121, 121, 121);
  border-right: 30px dashed rgb(121, 121, 121);

  padding: 20px;
  margin-top: 0px;
}

```

Látni, hogy a szaggatott vonalat a „**dashed**” kulcsszóval lehet létrehozni. Minden másra már volt előzőleg példa.

7. Az előző feladatsorhoz hasonlóan helyezzük el a képet a megfelelő helyre:

```

<body>
  <div class="content">
    <h1>
      
    </h1>

    <p>Programozási nyelvek</p>
    ....
    <p>Az olvashatóság számít.</p>
  </div>
</body>

```

Arra figyeljünk, hogy most a „**content**” osztályú (class) div-en belülre kerüljön a kép, majd a **<h1>** elemen belül található elemeket, és szövegeket rendeztessük középre:

```

h1 {
  text-align: center;
}

```

8. Csak gépeljük be a „C++ | Java | Python” szöveget az előző feladatban hozzáadott html kód utáni sorokba:

```

<h1>
  
</h1>

<p>C++ | Java | Python</p>

<p>Programozási nyelvek</p>

```

8., 9., 10., 11. Az előző feladatban begépelte paragrafusnak adjunk egy „**class**” attribútumot, akár le is cserélhetjük például egy `<div>` elemre:

```

<div class="menu">C++ | Java | Python</div>

```

Készítsük el hozzá a szükséges css-t:

```

.menu {
  padding: 5px;
  background-color: rgb(216, 216, 216);
  margin-top: 30px;
  margin-bottom: 30px;
  text-align: center;
}

```

12. Tegyük az alcímeket kettes szintű címsorrá (`<h2>`):

```

<body>
  <div class="content">
    ...
    <h2>Programozási nyelvek</h2>
    ...
    <h2>Magas szintű programozási nyelvek</h2>
    ...
  </div>
</body>

```

Majd készítsük el hozzá a szükséges css-t:

```

h2 {
  font-size: 22px;
  border-left: 2px solid black;
  border-bottom: 2px solid black;
  padding: 4px;
}

```

13., 15. Tegyük az alfejezeteket hármas szintű címsorrá (`<h3>`):

```

<body>
  <div class="content">
    ...
    <h3>C++</h3>
    ...
    <h3>Java</h3>
    ...
    <h3>Python</h3>
    ...
  </div>
</body>

```

Majd készítsük el hozzá a szükséges css-t:

```

h3 {
  font-size: 16px;
  border-bottom: 2px dotted black;
  padding: 2px;
}

```

A pontozott szegélyt a „**dotted**” kulcsszóval lehet elkészíteni.

14. A css-el úgy tudunk kiválasztani bizonyos elemeken belüli elemeket, hogy egymás után szünettel felsoroljuk őket. Tehát, ha a `<h3>` elemen belüli `<a>` elemeket szeretnénk megcélózni, akkor azt így tudjuk megtenni:

```

h3 a {
  font-size: 16px;
}

```

Megj.: Ez „**class**” attribútumokkal is működik, tehát ha mondjuk az összes `<div>` elemen belüli „**tartalom**” „**class**” attribútumú elemnek szeretnénk tulajdonságokat állítani, akkor ezt így tehetjük meg: „**div .tartalom { ... }**”

16. Tegyük a kért szövegeket négyes szintű címsorrá (**<h4>**):

```
<body>
  <div class="content">
    ...
    <h3>C++</h3>
    ...
    <h4>Filozófia</h4>
    ...
    <h3>Java</h3>
    ...
    <h4>Filozófia</h4>
    ...
    <h3>Python</h3>
    ...
    <h4>Filozófia</h4>
    ...
  </div>
</body>
```

Majd készítsük el hozzá a szükséges css-t:

```
h4 {
  font-size: 14px;
  color: #000b1a;
}
```

17. Helyezzük el a képeket a html dokumentumban a megfelelő helyre, és állítsunk be nekik egy új (pl. **kep\_jobb\_stilus**) „class” attribútumot:

```
<body>
  <div class="content">
    ...
    <h3>C++</h3>
    
    ...
    <h3>Java</h3>
    
    ...
    <h3>Python</h3>
    
    ...
  </div>
</body>
```

Majd készítsük el hozzá a szükséges css-t:

```
.kep_jobb_stilus {
  margin: 10px;
  padding: 5px;
  float: right;
  border: 1px solid rgb(0, 0, 0);
}
```

18. Első lépésként keressük meg az alcímeket, és tegyük linkké őket az `<a>` elemmel, illetve adjunk nekik nevet a „**name**” attribútummal:

```
<body>
  <div class="content">
    ...
    <h3><a name="cpp">C++</a></h3>
    ...
    <h3><a name="java">Java</a></h3>
    ...
    <h3><a name="python">Python</a></h3>
    ...
  </div>
</body>
```

Második lépésként alakítsuk linkeké a menü szavait, és a „**href**” attribútummal kapcsoljuk össze a nevekkel:

```
<body>
  <div class="content">
    <h1>
      
    </h1>

    <div class="menu">
      <a href="#cpp">C++</a> |
      <a href="#java">Java</a> |
      <a href="#python">Python</a>
    </div>

    <p>Programozási nyelvek</p>
```

19. Alakítsuk ki a kért táblázatot a szövegből a `<table>` elemmel. Az előző feladatsorhoz képes a különbség, hogy itt szükséges használnunk a `<thead>` elemet is a táblázat első sorának egyszerű kiemeléséhez:

```
<table align="center" border="1" width="600">
  <thead>
    <tr>
      <th>A Java nyelv létrehozásának öt fő célja volt</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>Egyszerűnek, és objektumorientáltnak kell lennie.</td>

    <tr>
      <td>Robusztusnak és biztonságosnak kell lennie.</td>
    </tr>
    <tr>
      <td>Architektúra-semlegesnek és hordozhatónak kell lennie.</td>
    </tr>
    <tr>
      <td>Nagy teljesítményűnek kell lennie.</td>
    </tr>
    <tr>
      <td>Értelmezettnek, szálakkal rendelkezőnek és dinamikusnak kell lennie.</td>
    </tr>
  </tbody>
</table>
```

Látni, hogy a `<thead>` elem ugyanúgy működik, mint a `<tbody>` elem.

Majd készítsük el hozzá a szükséges css-t:

```
table {
  border: 2px solid rgb(0, 0, 0);
  text-align: left;
}

thead {
  background-color: rgb(216, 216, 216);
}
```

20. Alakítsuk ki a számozott felsorolást:

```
<ol>
  <li>A szép jobb, mint a csúnya.</li>
  <li>Az explicit jobb, mint az implicit.</li>
  <li>Az egyszerű jobb, mint a bonyolult.</li>
  <li>Az összetett jobb, mint a bonyolult.</li>
  <li>Az olvashatóság számít.</li>
</ol>
```

Majd alakítsuk ki a számozatlan felsorolást:



```

<ul>
<li>A programnak tényleges problémákon kell alapulnia, és funkcióinak azonnal hasznosnak kell lenniük a valós világ programjaiban.</li>
<li>Minden funkciónak implementálhatónak kell lennie (ésszerűen nyilvánvaló módon).</li>
<li>A programozók szabadon választhatják meg a saját programozási stílusukat, és ezt a stílust a C++-nak teljes mértékben támogatnia kell.</li>
<li>Egy hasznos funkció engedélyezése fontosabb, mint a C++ minden lehetséges visszaélésének megakadályozása.</li>
<li>Lehetőségeket kell biztosítani a programok különböző, jól definiált részekre való szervezésére, és lehetőségeket kell biztosítani a különállóan fejlesztett részek kombinálására.</li>
<li>A típusrendszer implicit megsértése tilos (de az explicit, azaz a programozó által kifejezetten kért megsértések megengedettek).</li>
<li>A felhasználó által létrehozott típusoknak ugyanolyan támogatással és teljesítményűnek kell lenniük, mint a beépített típusoknak.</li>
<li>A nem használt funkciók nem lehetnek negatív hatással a létrehozott futtatható programokra (pl. alacsonyabb teljesítményben).</li>
<li>A C++ alatt nem lehet nyelv (kivéve az assembly nyelvet).</li>
<li>A C++-nak más létező programozási nyelvek mellett kell működnie, ahelyett, hogy saját különálló és inkompatibilis programozási környezetet alakítana ki.</li>
<li>Ha a programozó szándéka ismeretlen, akkor a programozó számára tegye lehetővé, hogy kézi vezérléssel adja meg azt.</li>
</ul>

```

21. A „content” „class” attribútumú `<div>` elem legaljára készítsünk egy linket, amely a „logok.html”-re mutat. Érdekes a link belső tartalmát egy div-be foglalni, és most nem „class” attribútumot, hanem „id” attribútumot fog kapni:

```

<body>
  <div class="content">
    ...

    <a href="logok.html">
      <div id="logo_link">Logók megtekintése nagyban</div>
    </a>
  </div>
</body>

```

Vigyázat! Két ugyanolyan „id” attribútumú elem nem lehet egy oldalon!

Majd készítsük el hozzá a szükséges css-t. A böngészőnek „#”-jellel kell jeleznünk, hogy „id” attribútumról van szó:

```

#logo_link {
  text-align: center;
  margin-top: 20px;
  margin-bottom: 20px;
  border-top: 2px dotted rgb(0, 0, 0);
  border-bottom: 2px dotted rgb(0, 0, 0);
  padding: 10px;
}

```

22. A feladat kérésének eleget téve alakítsuk ki a „logok.html”-t a „programozasi\_nyelvek.html”-ből, majd adjuk hozzá a kért képeket egy új „logok” „class” attribútumú <div>-be:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Programozási nyelvek</title>

  <style>
    ...
  </style>
</head>

<body>
  <div class="content">
    <h1>
      
    </h1>

    <h2>Logók</h2>

    <div class="logok">
      
      
      
    </div>
  </div>
</body>

</html>
```

Majd készítsük el hozzá a szükséges css-t:

```
.logok {
  text-align: center;
  margin: 50px;
}
```