

JavaScript alapok és adatvizualizáció p5.js-sel

MOME Open adatvizualizáció, 2022.

Dévai Nándor mail@nandordevai.net

JavaScript, HTML, CSS alapok

Használt eszközök

- VS Code
- Live Server plugin
- Git (Git Graph plugin)
- Web inspector/Developer tools/Web developer tools (JS konzol)

Példaprojektek

- <https://github.com/nandordevai/Dataviz-2022>

HTML és CSS alapok

HTML DOM

“The Document Object Model (DOM) is the data representation of the objects that comprise the structure and content of a document on the web.”

HTML és CSS alapok

HTML DOM

“The Document Object Model (DOM) is the **data representation** of the **objects** that comprise the **structure and content** of a document on the web.”

HTML és CSS alapok

CSS

- Class: DOM elemekhez rendelhető (nem egyedi) azonosító
- Selector: elemeket azonosító szabály
- Property: adott elem megjelenését befolyásoló tulajdonság

HTML és CSS alapok

Canvas

- HTML elem, amire JavaScripttel rajzolni lehet
- Diagramok, animációk létrehozására, ill. fotó vagy videó megjelenítésére és manipulációjára alkalmas
- Közvetlenül is scriptelhető, de p5.js-sel fogjuk használni

JavaScript alapok

Változók

- `const`: konstans érték, nem változtatható és nem deklarálható újra, de az objektum vagy tömb értékei módosíthatók
- `let`: változó, a konstans megkötései nélkül
- `block scope`

JavaScript alapok

Feltételes elágazás

- if ... else if ... else ...
- ternary operator: expression ? true : false
- switch

JavaScript alapok

Ciklusok

- „hagyományos” for ciklus – általában nem ez a jó választás
- forEach
- for ... of (iterable) és for ... in (enumerable)
- map(), reduce()
- Object.keys(), Object.values(), Object.entries()

JavaScript alapok

Függvények

- „first-class objects” – minden függvény egyben objektum is (Function object)
- függvénydefiniálás módjai
 - function statement
 - function expression (anonymous/named)
 - arrow function
 - method definition

JavaScript alapok

Osztályok

- egyedi objektumfajták létrehozására használjuk
- class utasítással definiáljuk
- constructor függvény inicializálja
- new utasítással hozunk létre új példányokat
- meglévő osztályokból származtathatunk újakat az extend utasítással

JavaScript alapok

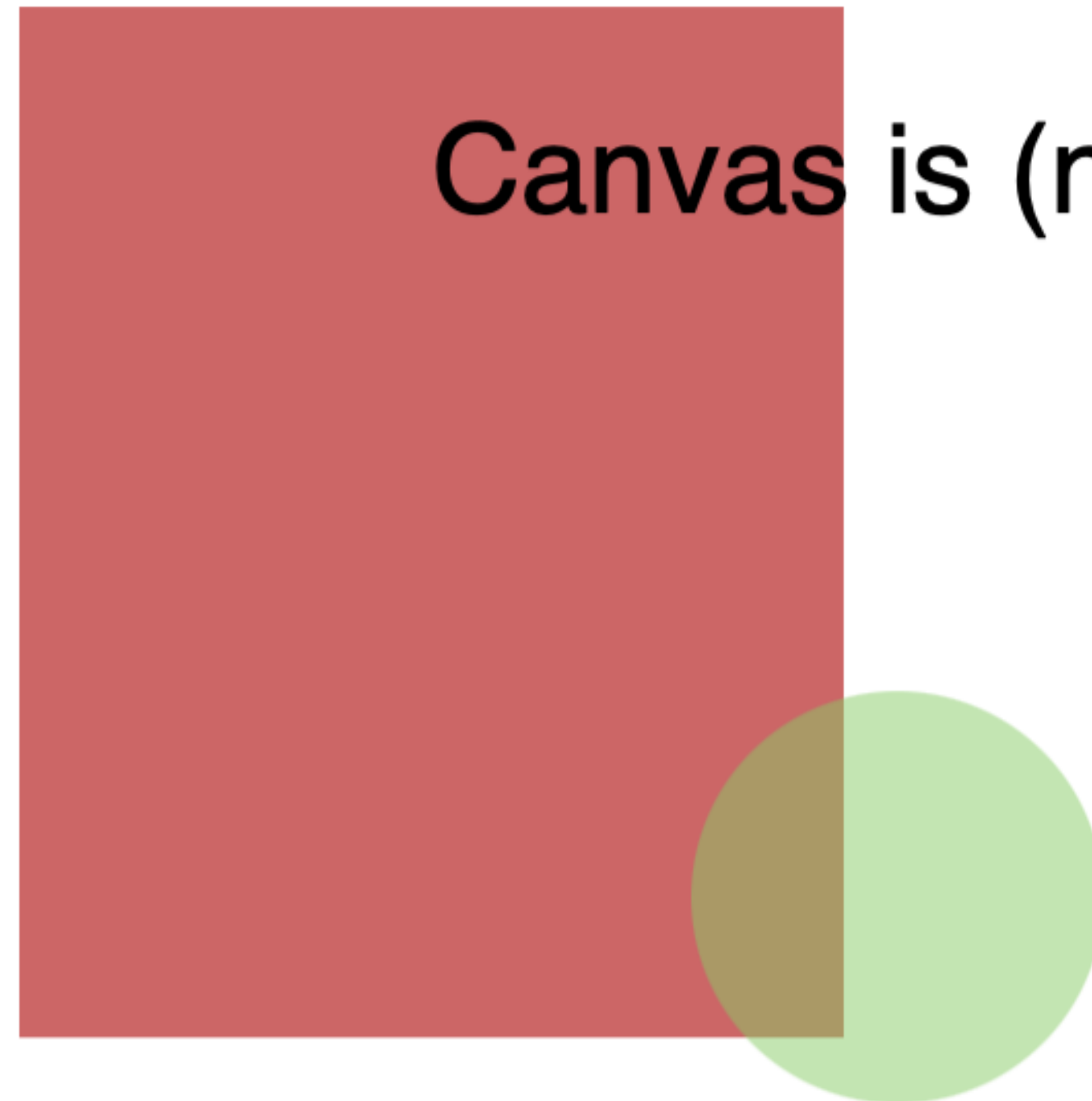
Modulok

- import utasítással történik
- az importáló fájlt modulként kell betölteni HTML-ből (script type="module")
- modulba kerülhet
 - osztály
 - utility függvények
 - config object
 - stb.

Demo

Canvas

Canvas is (not) fun



Adatvizualizáció p5.js-sel

p5.js

Alapfunkciók

- `setup()`
- `draw()`
- `loop()`, `noLoop()`, `redraw()`
- `createCanvas()`

p5.js

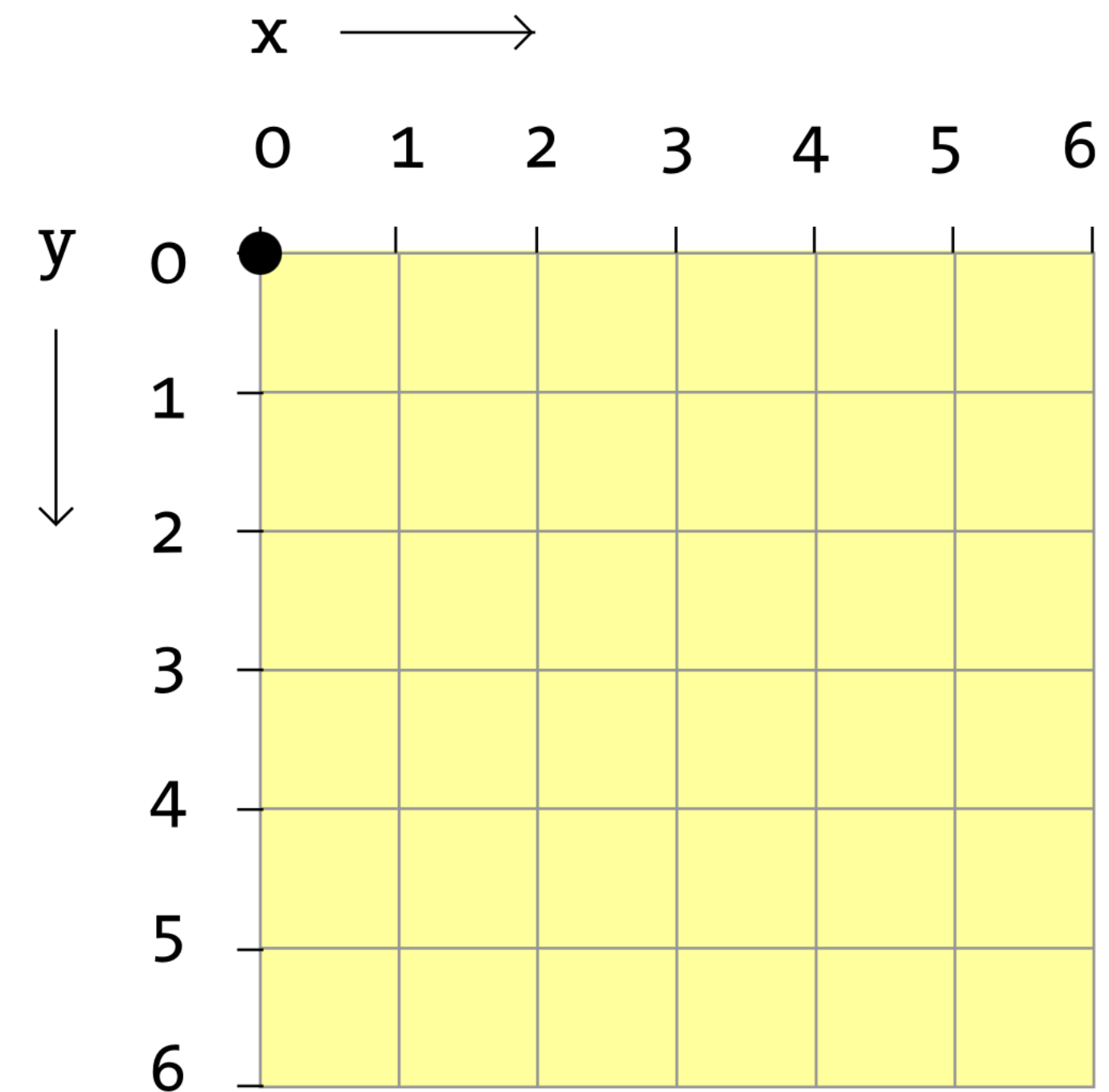
Objektumok

- 2D és 3D primitívek
- görbék (curve és bezier)
- vertex

p5.js

Koordináták és transzformáció

- origó a canvas bal felső sarkában, jobbra és lefelé pozitív
- `translate()`, `rotate()`, `scale()`
- `push()` és `pop()` függvényekkel elmenthetjük ill. visszaállíthatjuk az aktuális stílust és transzformációt



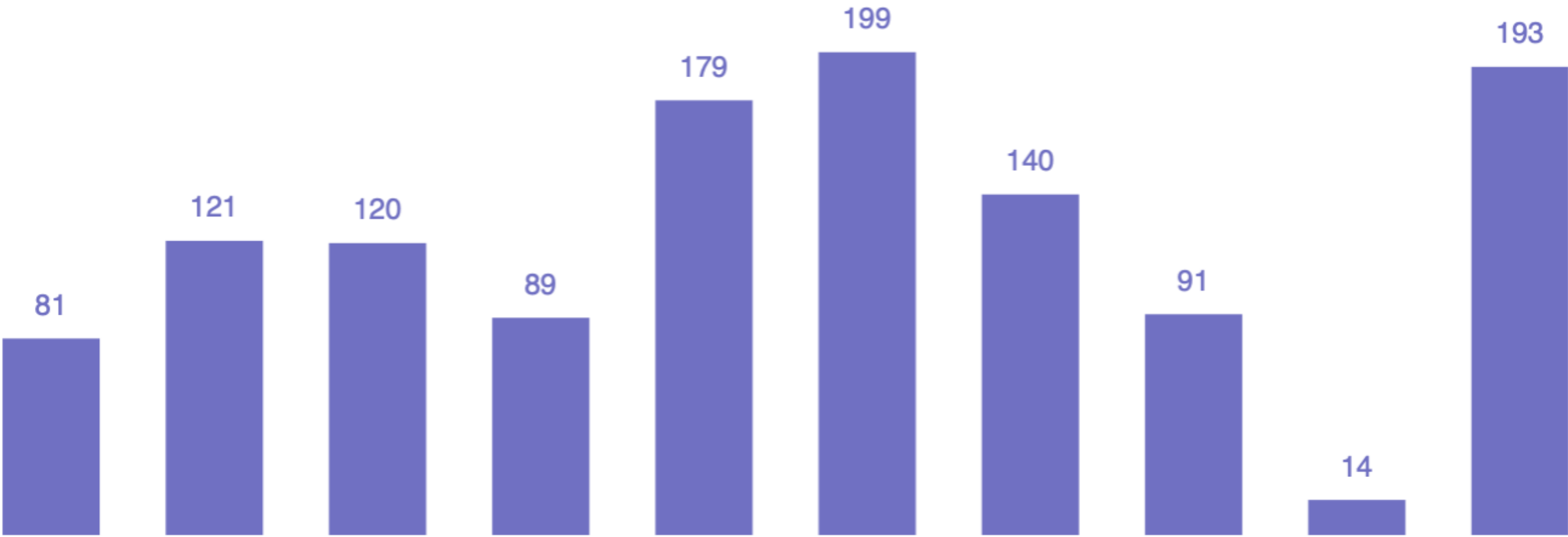
p5.js

Attribútumok

- color
 - colorMode(HSL) ajánlott
 - használható RGB, hex RGB, grayscale, color string
 - lerpColor()
- fill
- stroke

Demo

p5 bar chart



p5.js

Eseménykezelés

- billentyűzet
 - keyPressed(), keyReleased()
- egér
 - mousePressed(), mouseClicked(), mouseMoved()
 - mouseX, mouseY
- giroszkóp
- touch

p5.js

Adatok és fájlok betöltése

- metódusok: loadJSON(), loadTable() (CSV), httpGet()
- aszinkron függvénykezelés:
 - callback function
 - betöltés preload() függvényen belül

p5.js

Math

- `abs()`, `floor()`, `max()`, `min()`, stb.
- `lerp()` (lineáris interpoláció)
- `map()` (értéket átalakít két tartomány között)
- vektorok
 - `createVector(x, y, [z])`
 - `set()`, `add()`, `sub()`, `mult()`, stb.
- `noise()` (1, 2, vagy 3 dimenzióban), `random()` (intervallum v. tömb)

p5.js

HTML és CSS

- létrehozás: `createElement()` (+`createXXX`)
- kiválasztás: `select()`, `selectAll()` (CSS selector)
- stílusok beállítása: `el.addClass()`, `el.removeClass()`, `el.toggleClass()`
+ CSS class
- módosítás: `html()`, `attribute()`, `value()`

p5.js

Global vs. instance mode

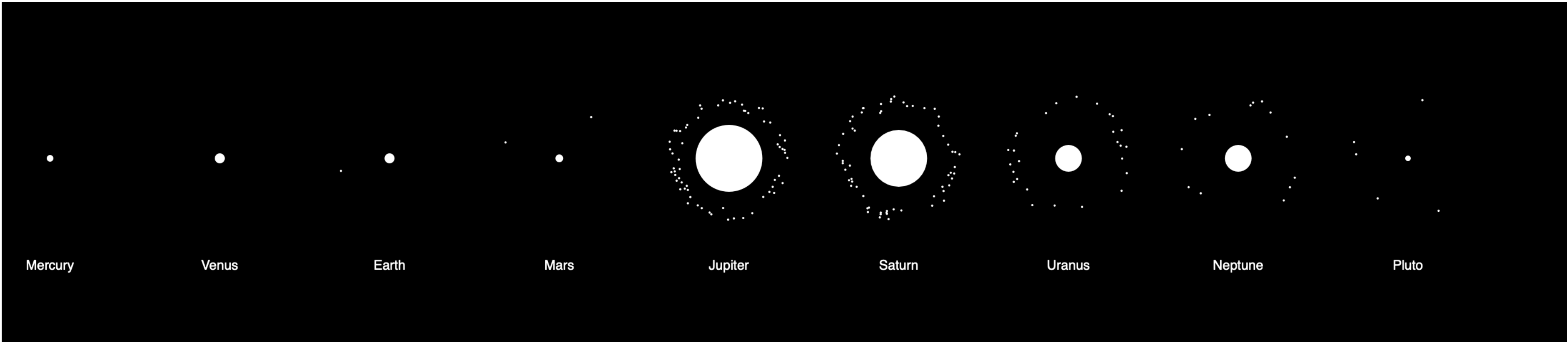
- alapértelmezett: global context (window object)
- más JS könyvtárakkal vagy egynél több sketch használata esetén problémát okozhat
- instance mode:

```
const sketch = (p5) => {  
  p5.setup = () => {...};  
  p5.draw = () => {...};  
};
```

```
const app = new p5(sketch, 'elementId');
```

Demo

Planets & moons



Források

Referenciák

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Reference>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>
- [https://developer.mozilla.org/en-US/docs/Web/API/Canvas API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API)
- <https://p5js.org/reference/>

Források

Könyvek, tutorialok, stb.

- The Nature of Code (<https://natureofcode.com>)
- The Coding Train (<https://www.youtube.com/c/TheCodingTrain/>)
- Generative Design: Visualize, Program, and Create with JavaScript in p5.js
- Processing Foundation (<https://discourse.processing.org/>)