

Wprowadzenie do p5.js

Krzysztof Gdawiec



UNIWERSYTET ŚLĄSKI
INSTYTUT INFORMATYKI

Tekst

p5.js wykorzystuje czcionki w formacie OpenType czyli *.otf*, *.ttf*.

Domyślnie w p5.js na starcie jest już załadowana pewna domyślna czcionka. Aby ustawić czcionkę według własnych potrzeb najpierw musimy ją załadować. Służy do tego funkcja

```
loadFont ( path )
```

`path` jest to ścieżka lub adres url do pliku z czcionką. Ścieżka powinna być względna (względem projektu p5.js). Ładowanie czcionki z adresu url może być blokowane przez ustawienia przeglądarki.

Funkcja `loadFont` działa w sposób asynchroniczny. Ponadto jako opcjonalne argumenty możemy podać funkcje zwrotne wykonywane po zakończeniu funkcji `loadFont` i w razie wystąpienia błędu (patrz dokumentacja).

Po wczytaniu czcionki musimy ustawić czcionkę jako bieżącą:

```
textFont( font )
```

```
textFont( font, size )
```

`font` to czcionka załadowana za pomocą `loadFont` lub string reprezentujący czcionkę, która dostępna jest w każdym systemie: https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals#Web_safe_fonts
`size` to rozmiar czcionki (w pikselach).

W celu wypisania tekstu korzystamy z funkcji:

```
text( str, x, y )
```

```
text( str, x, y, x2, y2 )
```

W obu wersjach `str` to napis jaki chcemy wypisać. Znaczenie pozostałych argumentów jest różne dla poszczególnych wersji funkcji `text`.

W pierwszej wersji `x` oznacza współrzędną X , od której rozpocznie się tekst, a `y` to współrzędna Y tzw. linii bazowej

Ściągając linia bazowa

W przypadku drugiej wersji funkcji `text` argumenty `x`, `y`, `x2`, `y2` wyznaczają prostokątny obszar dla tekstu. Interpretacja tych argumentów zależy od ustawionego za pomocą funkcji `rectMode` trybu rysowania prostokątów.

Do zmiany koloru czcionki możemy wykorzystać funkcję `fill`.

Do zmiany koloru obramowania czcionki możemy wykorzystać funkcję `stroke`, zaś do zmiany grubości obramowania funkcję `strokeWeight`.

```
1  function setup()
2  {
3    createCanvas( 600, 200 );
4
5    textFont( "Georgia", 50 );
6  }
7
8  function draw()
9  {
10   background( 0 );
11
12   fill( 0, 255, 0 );
13   stroke( 255, 0, 0 );
14   strokeWeight( 3 );
15
16   text( "Animacja_
17         komputerowa", 10, 100 );
```

Animacja komputerowa

Wielkość czcionki nie tylko możemy określić podczas ustawiania czcionki, ale również w dowolnym momencie działania programu. Służy do tego funkcja:

```
textSize( size )
```

`size` to rozmiar czcionki (w pikselach) jaki chcemy ustawić.

Mamy też możliwość zmiany sposobu wyrównania tekstu:

```
textAlign( hAlign )
```

```
textAlign( hAlign, vAlign )
```

`hAlign` ustawia wyrównanie w poziomie: `LEFT` (domyślne), `CENTER`, `RIGHT`.

`vAlign` ustawia wyrównanie w pionie: `TOP`, `BOTTOM`, `CENTER`, `BASELINE`.

Możemy również zmienić styl czcionki. Do tego celu służy funkcja:

```
textStyle( style )
```

`style` to styl jaki chcemy ustawić: `NORMAL`, `ITALIC`, `BOLD`, `BOLDITALIC`.

W p5.js mamy jeszcze kilka funkcji użytecznych funkcji związanych z tekstem. Pierwszą taką funkcją jest:

```
textWidth( str )
```

Funkcja zwraca szerokość podanego stringu `str` uwzględniając wielkość ustawionej czcionki.

W typografii istnieją różne metryki opisujące tekst. Dwoma najważniejszymi są: wydłużenie górne (ang. ascent) i wydłużenie dolne (ang. descent).



Pomimo tego, że w typografii istnieje więcej rodzajów metryk, to p5.js korzysta tylko z dwóch wymienionych na slajdzie metryk.

Do odczytania obu metryk w p5.js istnieją odpowiednie funkcje:

- ▶ wydłużenie górne

```
textAscent ()
```

- ▶ wydłużenie dolne

```
textDescent ()
```

Używając tych dwóch metryk jesteśmy w stanie np. obliczyć wysokość linii tekstu (suma obu wielkości). p5.js obie te metryki wykorzystuje przy wyrównaniu tekstu.

Obrazy

Stworzenie pustego obrazu:

```
createImage( w, h )
```

gdzie *w* to szerokość, *h* wysokość obrazu.

Funkcja wczytująca obraz z pliku przedstawia się następująco:

```
loadImage( filename )
```

gdzie *filename* to ścieżka (względem pliku HTML używającego p5.js) do pliku z obrazem lub adres internetowy. W przypadku adresu internetowego wczytywanie może zostać zablokowane przez ustawienia przeglądarki.

Wczytywany obraz może nie być natychmiast dostępny do dalszego przetwarzania lub renderingu. Jeśli chcemy mieć pewność, że obraz będzie wczytany przed jego użyciem, to wczytywanie takiego obrazu musimy umieścić w funkcji `preload`.

Mając wczytany/utworzony obraz możemy pobrać jego wymiary odwołując się do pól: `width` (szerokość), `height` (wysokość).

Kiedy mamy już wczytany/stworzony obraz możemy wyświetlić go wykorzystując funkcję:

```
image( img, x, y )
```

```
image( img, x, y, w, h )
```

Argument `img` jest to obraz, który chcemy wyświetlić w oknie programu.

Pierwsza wersja funkcji jest skróconą wersją drugiej, gdzie przyjęto, że `w` równe jest szerokości obrazu, a `h` jego wysokości.

Interpretacja argumentów funkcji `image` zależy od ustawionego trybu, który możemy ustawić za pomocą funkcji:

```
imageMode( mode )
```

gdzie argument `mode` wyznacza tryb, w którym chcemy pracować. Przyjmować może on wartości: `CORNER` (domyślny), `CORNERS`, `CENTER`. Znaczenie poszczególnych trybów jest identyczne jak w przypadku funkcji `rectMode`.

```
1  let img;
2
3  function preload()
4  {
5    img = loadImage( "sunflower.jpg" );
6  }
7
8  function setup()
9  {
10   createCanvas( 400, 400 );
11 }
12
13 function draw()
14 {
15   background( 0 );
16
17   imageMode( CORNER );
18   image( img, 0, 0 );
19
20   imageMode( CENTER );
21   image( img, img.width + 60, 200, 100, 100
22         );
23 }
```



W pracy z obrazami bardzo często przydaje się pobranie lub ustawienie koloru określonego piksela. W p5.js istnieją dwa sposoby umożliwiające wykonania tych zadań.

Pierwszym sposobem jest użycie odpowiednich metod.

- ▶ Metoda zwracająca kolor lub wycinek obrazu

```
get ()
```

```
get ( x, y )
```

```
get ( x, y, w, h )
```

Pierwsza metoda zwraca cały obraz, druga kolor piksela o współrzędnych x , y . Ostatnia metoda zwraca wycinek obrazu zdefiniowanego przez: x , y – współrzędne lewego górnego rogu, w – szerokość, h – wysokość.

- ▶ Metoda ustawiająca kolor piksela lub fragmentu obrazu

```
set( x, y, c )
```

Metoda ustawia piksel o współrzędnych x , y na kolor c przy czym c może być tablicą pikseli.

Operacje, które umożliwiają metody `get` i `set` możemy również przeprowadzić na zawartości okna programu. Funkcje, które realizują te zadania mają dokładnie takie same nazwy i przyjmują takie same argumenty co metody `get` i `set`.

Użycie metod/funkcji `get` i `set` jest wygodne, ale przy dużej liczbie ich wywołań, co ma np. miejsce przy przetwarzaniu obrazu, program może drastycznie zwolnić. Dlatego też w `p5.js` mamy drugi sposób odwoływania się do pikseli obrazu.

Obiekt obrazu ma pole o nazwie `pixels`. Jest to tablica jednowymiarowa zawierająca piksele obrazu (4 kolejne elementy tej tablicy to kanały R, G, B, A piksela). Zanim jednak odwołamy się do tej tablicy musimy ją zapełnić aktualnym stanem pikseli. W tym celu wywołujemy metodę:

```
loadPixels()
```

Po skończonej pracy na tablicy `pixels` musimy przenieść dane z tej tablicy do obrazu. Aby wykonać tę czynność wywołujemy metodę:

```
updatePixels()
```

```
1  let img;
2
3  function preload() {
4    img = loadImage( "sunflower.jpg" );
5  }
6  function setup() {
7    createCanvas( 528, 400 );
8  }
9  function draw() {
10   background( 0 );
11   image( img, 0, 0 );
12
13   let image_size = img.width * img.height;
14   let form_size = width * height;
15
16   loadPixels();
17   img.loadPixels();
18   for( let i = 0; i < image_size; ++i ) {
19     let index = form_size - ( floor( i / img.width
20       ) * width + ( i % img.width ) + 1 );
21
22     pixels[4 * index] = img.pixels[4 * i];
23     pixels[4 * index + 1] = img.pixels[4 * i + 1];
24     pixels[4 * index + 2] = img.pixels[4 * i + 2];
25     pixels[4 * index + 3] = img.pixels[4 * i + 3];
26   }
27   img.updatePixels();
28   updatePixels();
29 }
```

