

## Materiały powtórzeniowe

### Obiekty



7 Witaj w materiałach powtórzeniowych przygotowanych specjalnie dla Ciebie. Znajdziesz tutaj powtórkę materiału z ostatniej lekcji oraz zadania do wykonania dla chętnych. Z pewnością sobie poradzisz. Powodzenia!



## Cel lekcji

Celem lekcji jest poznanie czym są i jak tworzy się obiekty w JavaScript.

Adres projektu <https://giganci.trinket.io/sites/obiekty-gp>

## Obiekty i właściwości

W świecie programowania każdą rzecz świata fizycznego możemy przedstawić w postaci obiektu. Przykładem na którym będziemy obrazować obiekty będzie motocykle. Każdy obiekt posiada swoje właściwości, zdarzenia oraz metody.

## Właściwości

Oba motocykle mają te same cechy charakterystyczne.

Każdy motocykl ma producenta, kolor, pojemność silnika. Takie cechy nazywane są **właściwościami**.

Każda taka właściwość ma **nazwę i wartość**, która dostarcza informacje o danym obiekcie



Taka idea par nazwa-wartość jest stosowana w HTML i CSS np.

`type="email"`

`color:#ffffff`

### Obiekt motocykl

Oba egzemplarze obiektu motocykl współdzielą te same właściwości ale wartości mogą być różne



Obiekt motocykl	
Właściwości	Wartość
producent	Yamaha
kolor	Niebieski
Prędkość maksymalna	120km/h





Obiekt motocykl	
Właściwości	Wartość
producent	Honda
kolor	Czarny
Prędkość maksymalna	250km/h

## Zdarzenia

Są to interakcje, które mogą spowodować zmianę wartości właściwości

Np. naciskając dźwignię hamulca zmniejsza się prędkość motocykla

Take zdarzenie mogłoby się nazywać hamuj

Programy komputerowe są zaprojektowane do wykonywania różnych rzeczy w zależności interakcji użytkownika z komputerem. Np. klikając łącze zaloguj, użytkownik przenoszony jest do formularza gdzie może wprowadzić dane i się zalogować



## Metody

Dostarczają informacji o obiekcie oraz mogą zmieniać wartość właściwości obiektu np. Metoda `zmienPredkosc()` mogłaby zwiększyć lub zmniejszyć prędkość motocykla

Zdarzenia i metody są ze sobą powiązane np.

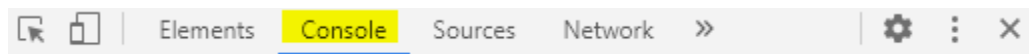
Kiedy motocyklista naciśnie kłamkę hamulca następuje wywołanie zdarzenia **hamuj**, które uruchamia metodę **zmienPredkosc()**, która to zmienia wartość właściwości `aktualnaPredkosc`.

**Przeglądarki internetowe** tworzą modele wyświetlanej strony w bardzo podobny sposób

Każde okno lub karta przeglądarki jest przedstawiona za pomocą obiektu **window**

Natomiast każda wczytana strona w oknie przeglądarki reprezentowana jest za pomocą obiektu **document**

Przechodzimy na dowolną stronę klikamy F12, przechodzimy na zakładkę Console



Wpisujemy `window.location` taki zapis wyświetli nam wartość właściwości `location` (która to również jest obiektem) obiektu `window`



```
> window.location
< Location {href: "https://www.giganciprogramowania.edu.pl/", ancestorOrigins: DOMStringList, origin: "https://www.giganciprogramowania.edu.pl", protocol: "https:", host: "www.giganciprogramowania.edu.pl", ...}
  ▶ ancestorOrigins: DOMStringList {length: 0}
  ▶ assign: f assign()
  ▶ fragmentDirective: FragmentDirective {}
    hash: ""
    host: "www.giganciprogramowania.edu.pl"
    hostname: "www.giganciprogramowania.edu.pl"
    href: "https://www.giganciprogramowania.edu.pl/"
    origin: "https://www.giganciprogramowania.edu.pl"
    pathname: "/"
    port: ""
    protocol: "https:"
  ▶ reload: f reload()
  ▶ replace: f replace()
    search: ""
  ▶ toString: f toString()
  ▶ valueOf: f valueOf()
    Symbol(Symbol.toPrimitive): undefined
  ▶ proto : Location
```

document.title właściwość title obiektu document zwróci nam nazwę strony która jest ustawiona pomiędzy znacznikami <title></title>

```
> document.title
< "Giganci Programowania! Nauka programowania dla dzieci i młodzieży"
> |
```

Oczywiście obiekt document posiada zdarzenia np.

click kiedy to klikniemy na jakiś element może zostać to wykryte  
load-kiedy nastąpi zakończenie wczytywania strony i zasobów



Obiekt document posiada również metody jedną z nich jest **querySelector()** którą bardziej poznamy na przyszłych zajęciach a teraz tylko wykonamy przykład.

Na stronie giganci programowania zmienimy kolor nagłówków, oczywiście zmiana ta będzie tylko lokalnie na naszej kopii strony wczytanej w przeglądarce.



## Tworzenie własnych obiektów-notacja Literału

Podsumowując obiekt to zgrupowany zbiór zmiennych i funkcji.

W obiekcie zmienne nazywać będziemy właściwościami

Funkcje która jest częścią obiektu nazywać będziemy metodami.

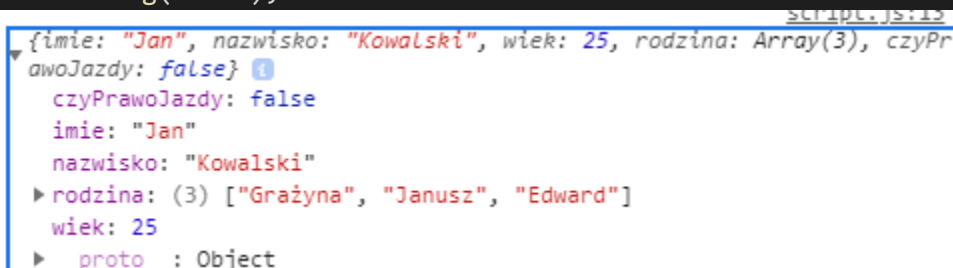
Notacja literału to najłatwiejszy i najpopularniejszy sposób tworzenia obiektów



Obiekt to w tym przypadku nawias klamrowy i jego zawartość

Klucz od wartości oddzielony jest dwukropkiem a poszczególne właściwości i metody są rozdzielone przecinkami.

```
const osoba={  
  
  /*klucz*/ imie:'Jan',/*wartość*/  
  nazwisko:'Kowalski',  
  wiek:25,  
  rodzina:['Grażyna','Janusz','Edward'],  
  czyPrawoJazdy:false  
  
};  
  
console.log(osoba);
```



```
SCRIPT100:JS:13  
{imie: "Jan", nazwisko: "Kowalski", wiek: 25, rodzina: Array(3), czyPr  
awoJazdy: false} 1  
  czyPrawoJazdy: false  
  imie: "Jan"  
  nazwisko: "Kowalski"  
  ▶ rodzina: (3) ["Grażyna", "Janusz", "Edward"]  
  wiek: 25  
  ▶ __proto__: Object
```

Widzimy że w obiekcie mogą występować różnego typu dane.

Aby uzyskać dostęp do właściwości stosujemy **operator elementu składowego-kropkę „.”**

```
console.log(osoba.wiek);
```

25



```
console.log(osoba.czyPrawoJazdy);
```

false

LUB poprzez nawiasy kwadratowe

```
console.log(osoba['wiek']);  
console.log(osoba['nazwisko']);
```

25

Kowalski

Wartości możemy edytować

```
osoba.wiek=40;  
osoba.nazwisko='Nowak';
```

```
console.log(osoba);
```

```
{  
  "imie": "Jan",  
  "nazwisko": "Nowak",  
  "wiek": 40,  
  "rodzina": Array(3) ["Grażyna", "Janusz", "Edward"],  
  "czyPrawoJazdy": false  
}
```

Teraz usuniemy w naszym obiekcie osoba właściwość wiek a zamiast tego dodamy funkcję która będzie ten wiek obliczać w zależności od podanego roku urodzenia



```
const osoba={  
  
  /*klucz*/ imie:'Jan',/*wartość*/  
  nazwisko:'Kowalski',  
  rodzina:['Grażyna','Janusz','Edward'],  
  czyPrawoJazdy:false,  
  obliczWiek:function(rokUrodzenia){  
    return 2022-rokUrodzenia;  
  }  
  
};  
  
console.log(osoba.obliczWiek(1993));
```

**this**

```
const osoba={  
  
  /*klucz*/ imie:'Jan',/*wartość*/  
  nazwisko:'Kowalski',  
  rodzina:['Grażyna','Janusz','Edward'],  
  czyPrawoJazdy:false,  
  rokUrodzenia:1993,  
  obliczWiek:function(rokUrodzenia){  
    return 2022-this.rokUrodzenia;  
  }  
  
};
```



Słowo kluczowe `this` wskazuje na ten obiekt czyli osoba a `osoba.rokUrodzenia` zwróci 1993

Możemy również użyć `this` do dodania nowej właściwości

```
const osoba={  
  
  /*klucz*/ imie:'Jan',/*wartość*/  
  nazwisko:'Kowalski',  
  rodzina:['Grażyna','Janusz','Edward'],  
  czyPrawoJazdy:false,  
  rokUrodzenia:1993,  
  obliczWiek:function(rokUrodzenia){  
    /*wskazujemy nową właściwość wiek*/    this.wiek = 2022-this.rokUrodzenia  
  }  
};
```



## Questy dla Programisty



Spróbuj swoich sił w Questach Programisty. Jest to dodatkowy trening na najwytrwalszych programistów. Pamiętaj jednak, że nie jest to praca domowa i nie jest wymagana na lekcji. Gotowi?

### Quest 1.

Stwórz obiekt auto do którego dodasz właściwości takie jak: marka, model, rocznik, kolor, cena. Do obiektu dodaj funkcję info() której zadaniem będzie wyświetlanie danych o obiekcie.

