

Najczęstsze polecenia

Tworzenie stron internetowych



Witaj w materiałach pomocniczych przygotowanych specjalnie dla Ciebie. Znajdziesz tutaj najczęstsze kody i komendy potrzebne przy programowaniu. Zapisz, wydrukuj i korzystaj! Powodzenia



JavaScript

Zmienne

Nowe podejście to używanie słów kluczowych **let** i **const**. Słowo **let** oznacza zmienną, natomiast **const** stałą (taką, do której po ustawieniu wartości, nie możemy przypisać nowej)

```
let wiek=15;
console.log(wiek);
wiek=14;
console.log(wiek);
```

```
const wiek=15;
console.log(wiek);
wiek=14; //tutaj otrzymamy błąd ponieważ chcemy zmienić wartość stałej
console.log(wiek);
```

Podstawowe typy danych - JavaScript rozróżnia liczby, ciągi tekstowe oraz wartości true i false- typy boolowskie

- **typy liczbowe** - przeznaczone do obsługi liczb

```
let liczba1=15;
console.log(liczba1);
liczba1=14.453; //części ułamkowe podajemy po kropce
console.log(liczba1);
liczba1=-13.2
console.log(liczba1);
```

- **Typy tekstowe**-dane w postaci ciągów tekstowych składających się z liter oraz innych znaków

```
const nazwaSzkoły='Giganci programowania';
console.log(nazwaSzkoły);
```



- **Typy boolowskie**-typ ten może przyjmować jedną z wartości true lub false

```
const czyWlaczony=false;  
console.log(czyWlaczony);
```

- **Undefined** – zmienna która nie ma jeszcze przypisanej wartości

```
let liczba;  
console.log(liczba);
```

Pobieranie danych od użytkownika

```
const odpowiedz=prompt('Ile masz lat?');  
console.log(odpowiedz);
```

Operator ciągu tekstowego +

Proces łączenia wielu ciągów znaków nazywamy konkatencją.

```
const nazwa1='Giganci';  
const nazwa2='Programowania'  
const nazwaFirmy=nazwa1+' '+nazwa2;  
alert(nazwaFirmy); //wyświetla dane w wyskakującym okienku przeglądarki
```

Operatory Arytmetyczne

Nazwa działania	Operator	Opis	Przykład	Wynik
Dodawanie	+	Dodanie jednej wartości do drugiej	12+3	15



Odejmowanie	-	Odjęcie jednej wartości od drugiej	5-2	3
Dzielenie	/	Podzielenie dwóch wartości	10/2	5
Mnożenie	*	Pomnożenie dwóch wartości	10*5	50
Inkrementacja	++	Dodanie wartości 1 do bieżącej	X=5; X++;	6
Dekrementacja	--	Odjęcie wartości 1 od bieżącej	i=10; i--;	9
Modulo	%	Podzielenie dwóch wartości i zwrot reszty z dzielenia	10%2	0

Uwaga operator inkrementacji i dekrementacji może występować przed lub po zmiennej.

Gdy występuje przed zmienną, zmiana jej wartości nastąpi w danej instrukcji. Jeżeli występuje po zmiennej, zmiana wartość wystąpi dopiero w kolejnej instrukcji.

```
let liczba=10;
console.log(--liczba)//wynik 9
console.log(liczba)//wynik 9
```

```
let liczba=10;
console.log(liczba--)//wynik 10
console.log(liczba)//wynik 9
```

Operatory przypisania- operatory, które służą do przypisania zmiennej jakiejś wartości.

Przykład dla x=4 i y=3

Operator	Przykład	Równoznaczne	Wynik
=	x=y	x=y	x=3
+=	x+=y	x=x+y	x=7



-=	x-=y	x=x-y	x=1
=	x=y	x=x*y	x=12
/=	x/=y	x=x/y	x=1.33
%=	x%=y	x=x%y	x=1

Przykład

```
let wiek=13;
wiek+=4;
console.log(wiek);//17
```

```
let tekst='Giganci programowania';
tekst+= ' to innowacyjna';
tekst+= ' szkoła ';
tekst+= 'programowania';
console.log(tekst);
```

Podejmowanie decyzji

W skrypcie znajduję się wiele miejsc, w których podejmowane są decyzje wpływające na wykonywanie dalszego kodu. Żeby można było podjąć odpowiednią decyzję będziemy potrzebować operatorów porównania.

Operatory porównania służą do porównywania wartości w skrypcie z oczekiwaną, wynikiem będzie wartość boolowska **true** lub **false**.

Operator	Opis	Równanie	Wynik boolowski
==	Równy z	4==4	true
!=	różny	3!=5	true
===	Identyczny z	'3'===3	false
!==	Nieidentyczny z	'3'!==3	false



>	Większy niż	5>2	true
<	Mniejszy niż	5<10	true
>=	Większy niż lub równy	3>=3	true
<=	Mniejszy niż lub równy	2<=3	true

Instrukcja warunkowa

```
const liczba1=10;
const liczba2=10;

if(liczba1>liczba2)//porównanie zwraca false
{
    console.log(liczba1+' jest większe od '+liczba2); //ten kod się nie wykonuje
}

console.log('Koniec działania programu');
```

Instrukcja If else

```
const liczba1=10;
const liczba2=10;

if(liczba1>liczba2)
{
    console.log(liczba1+' jest większe od '+liczba2);
}
else{
    console.log(liczba1+' jest mniejsze lub równe '+liczba2);
}

console.log('Koniec działania programu');
```



If else if else

```
const x = 0;
if(x < 0){
  alert("x jest ujemne");
}
else if (x == 0) {
  alert("x jest równe 0");
}
else {
  alert("x jest większe od 0");
}
```

Operatory logiczne- Są one także stosowane w instrukcjach warunkowych. Za ich pomocą możemy łączyć kilka porównań w jedną całość.

Operator	Opis	Przykład	Wynik
&&	and(i)	(5>3)&&(4<7)	true
	or(lub)	(8==7) (9<10)	true
!	negacja	!(5==5)	false

Switch

```
const ocena=prompt('Podaj ocenę 1-5');

switch(ocena){

case '5':
  alert('Bardzo dobry');
  break;

case '4':
```



```
    alert('Dobry');  
    break;  
  
case '3':  
    alert('Dostateczny');  
    break;  
  
case '2':  
    alert('Dopuszczający');  
    break;  
  
case '1':  
    alert('Niedostateczny');  
    break;  
  
default:  
    alert('Podałś złą wartość');  
}
```

Funkcje

```
function powitanie(){// deklaracja funkcji  
  
const imie=prompt('Jak masz na imię?');  
alert('Witaj! ' + imie + ' na mojej stronie');  
  
}  
  
powitanie();//wywołanie funkcji  
powitanie()
```

Parametry funkcji

```
function wiekUczestnika(rokUrodzenia){  
  
    return 2020 - rokUrodzenia;  
}
```



```
}  
  
const wiekAdama=wiekUczestnika(2000);  
const wiekDarka=wiekUczestnika(2004);  
const wiekPawla=wiekUczestnika(2006);  
  
console.log(wiekAdama,wiekDarka,wiekPawla);
```

Tablice

```
const imiona=['Adam','Paweł','Tomek','Darek'];  
const liczby=[1,4,65,34];  
const tablica=['kokatenacja','javascript',2,5,78.65]//różny typ danych
```

Indeks 0	Indeks 1	Indeks 2	Indeks 3
Adam	Paweł	Tomek	Darek
1	4	65	43

Odwoływanie się do elementów tablicy

```
console.log(imiona[2]);//Tomek  
console.log(liczby);
```

Właściwość length

```
console.log(imiona.length);//4
```

Dodawanie elementów do tablicy

push(element1,element2)

```
imiona.push('Czesiek');
```



```
console.log(imiona);
```

Usuwanie ostatniego elementu tablicy
pop()

```
const liczby=[1,4,65,34,-43];  
console.log(liczby.pop()); //-43
```

Usuwanie pierwszego elementu tablicy
shift()-usuwa pierwszy element z tablicy i zwraca jego wartość

```
const imiona=['Adam','Pawel','Tomek','Darek'];  
imiona.shift();//Adam  
console.log(imiona);Pawel Tomek Darek
```

Łączenie elementów tablicy

```
console.log(imiona.join(' - ')); //Pawel - Tomek - Darek - Czesiek
```

Pętle

Pętla for

```
for (let i=0; /*licznik pętli*/ i<10; /*warunek kończący wykonywanie pętli;*/ i++  
/*zmiana wartości licznika*/) {  
    // kod który zostanie wykonany pewną ilość razy  
}
```

```
for(let i=1;i<=100;i++){  
    console.log('Giganci programowania');  
}
```



Pętla while

```
let i=5;
while (i<=10 /*wyrażenie sprawdzające zakończenie pętli*/) {
  //fragment kodu który będzie powtarzany
}
```

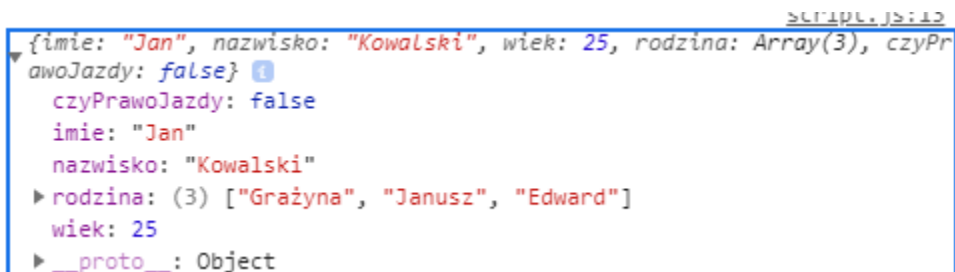
Obiekty

```
const osoba={

  /*klucz*/ imie:'Jan',/*wartość*/
  nazwisko:'Kowalski',
  wiek:25,
  rodzina:['Grażyna','Janusz','Edward'],
  czyPrawoJazdy:false
};

console.log(osoba);
```

Wynik w konsoli:



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

Odwołanie do właściwości obiektu

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



DOM – obiektowy model dokumentu

Odwoływanie się do elementów strony

`getElementById()`

```
<p id="tekst">
    JavaScript
</p>

<script>
    document.getElementById('tekst').textContent='Giganci programowania';
</script>
```

Dodanie nowego kodu HTML

```
<script>
document.getElementById('tekst').innerHTML='<h1> Giganci programowania &#10100;&#10101;</h1>';
</script>
```

`querySelector()`

```
<div id="zdjecie">
    
</div>
```



```
document.querySelector('#zdjecie img').addEventListener('click',function(){  
//właściwości css które składają się z wielu wyrazów tutaj zapisujemy w notacji  
camelCase np. borderRadius  
    document.querySelector('img').style.borderRadius='50px';  
});
```

W metodzie **querySelector()** rzeczywiście możemy stosować selektory CSS i odwoływać się w różny sposób do danego elementu

```
document.querySelector('#zdzj').addEventListener('click',function(){  
    document.querySelector('img').style.borderRadius='50px';  
});
```

lub

```
document.querySelector('.zdzj').addEventListener('click',function(){  
    document.querySelector('img').style.borderRadius='50px';  
});
```

lub

```
document.querySelector('.zdzj').addEventListener('click',function(){  
    document.querySelector('img').style.borderRadius='50px';  
});
```

lub

```
document.querySelector('div img').addEventListener('click',function(){  
    document.querySelector('img').style.borderRadius='50px';  
});
```



Obserwator zdarzeń

addEventListener()

```
<p id="doTlumaczenia">
```

Najczęściej spotykanym zastosowaniem języka JavaScript są strony internetowe. Skrypty te służą najczęściej do zapewnienia interakcji poprzez reagowanie na zdarzenia, walidacji danych wprowadzanych w formularzach lub tworzenia złożonych efektów wizualnych.

```
</p>
```

```
<button id="btnTlumacz">
```

Tłumacz tekst

```
</button>
```

```
//wywołujemy metodę addEventListener na wskazanym elemencie jako pierwszy argumen  
t podajemy rodzaj zdarzenia jako drugi nazwę funkcji do wywołania, w naszym przyp  
adku funkcja jest bez nazwy tzw funkcja anonimowa
```

```
document.getElementById('btnTlumacz').addEventListener('click',function(){
```

```
document.getElementById("doTlumaczenia").textContent="The most common Jav  
aScript language are websites. These scripts are most often used to connect by re  
sponding to events, validating data entered in the form, or creating complex visu  
al effects.";
```

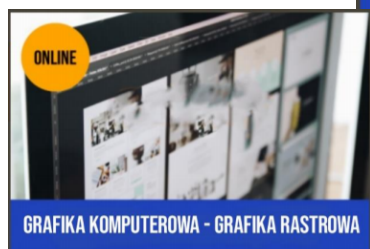
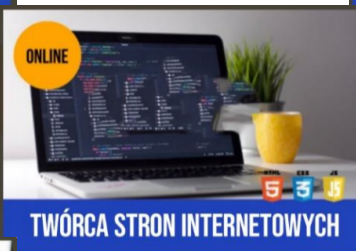
```
});
```







Chcesz stworzyć jeszcze więcej gier i wypróbować nowe możliwości? Zapraszam Cię na krótkie kursy online, w których możesz uczestniczyć w dowolnym czasie. Nowe gry, kolejne wyzwania - dla każdego, kto w grach i programowaniu chce być mistrzem. Zajrzyj, sprawdź, skorzystaj!



Zapisz się
już dziś!





<https://www.giganciprogramowania.edu.pl/>

sekretariat@giganciprogramowania.edu.pl

tel: 22 112 10 63

