

# Javascript fogalomtár

## **Szintaktika:**

az adott programozási nyelv nyelvtana, szintaktikailag helyes valami, ha helyesen van leírva.

## **Szemantika:**

a jelentés, szemantikailag helyes valami, ha helyesen működik.

## **Compileres nyelv:**

A megírt programkódot először egy compiler gépi kóddá alakítja (fordítja), ezután kerül futtatásra.

## **Interpreteres nyelv:**

A megírt program rögtön futtatható, nem kell fordítani.

## **Compiler:**

Egy program, amely valamilyen programozási nyelven megírt programot gépi kóddá fordít.

## **Interpreter:**

Egy program, amely valamilyen programozási nyelven megírt programot utasításonként értelmez és végre is hajt.

## **Case-sensitive:**

A case-sensitive nyelvek megkülönböztetik a kisbetűket és a nagybetűket.

## **Case-insensitive:**

A case-insensitive nyelvek nem különböztetik meg a kisbetűket és a nagybetűket.

## **Foglalt szavak:**

A nyelv rögzített jelentéssel bíró szavai, amelyet más célra nem lehet használni.

## **Változó:**

A változó egy elnevezett memória terület. Valamilyen érték tárolására használjuk, melyet a program futtatása során bármikor megváltoztathatunk.

## **Deklarálás:**

Egy változó nevének (egyes programozási nyelvekben a típusának is) a megadása. Azt fejezzük ki, hogy megadunk a fordító számára egy használni kívánt változó nevet és típust.

## **Definiálás:**

Ha változó definiálását emlegetünk változó deklaráció helyett, akkor azzal azt hangsúlyozzuk ki, hogy lefoglalódik egy hely a változónak a memóriában.

## **Inicializálás:**

Inicializálásnak nevezzük, amikor egy változó definiálásával egyidejűleg értéket is adunk a változónak.

## **Kifejezés:**

Kifejezés elemek összekapcsolva operátorokkal.

## **Kifejezés kiértékelés:**

Ha egy kifejezés több operátort is tartalmaz, akkor a kiértékelésük sorrendjét a precedencia szabályok határozzák meg. A zárójellezett kifejezésben szereplő operátorok a teljes kifejezés kiértékelése szempontjából nem számítanak, hiszen a zárójellezett kifejezés külön értékelődik ki.

## **Precedencia szintek:**

Minden operátor be van sorolva valamilyen precedencia szintre. Egy precedencia szinten lehet több operátor is.

## **Precedencia szabályok:**

A különböző precedencia szintű operátorok esetén mindig a magasabb precedenciájú hajtódik végre először. Az azonos precedencia szintű operátorok közül a leírás sorrendisége dönt. A legtöbb precedencia szint esetén a leírás sorrendjében, azaz balról-jobbra hajtódnak végre. Egyes precedencia szinteknél előfordul, hogy jobbról-balra.

**Szelekció:**

Választás megadott tevékenységek közül.

**Szekvencia:**

Egymás után végrehajtható utasítások sorozata.

**Iteráció:**

Megadott tevékenységek ismételt végrehajtása.

**Függvény:**

Olyan utasításcsoportok, amelyek segítségével egy-egy utasításcsoport a program többi részétől elkülöníthető, névvel látható el, és a program további részében e névvel lehet rá hivatkozni.

**Visszatérési érték:**

Általában a függvények eredménye egyetlen jól körülhatárolt érték, adat, vagy objektum, amit a függvény visszatérési értékének nevezünk.

**Paraméter:**

A függvény eredményének képzéséhez esetleg szükséges paraméterek, azaz adatok, melyeket argumentumoknak is hívjuk.

**Globális változó:**

A programban bárhol elérhető változók. Emiatt a hatókörük globális.

**Lokális változó:**

A függvényeken belül deklarált változók, melyeket csak az adott függvényen belül érhetünk el, használhatunk.

**Objektum:**

Információt (adatokat) tárol és kérésre műveleteket végez. Van állapota, viselkedése és futásidőben azonosítható.

**Metódus:**

Az objektumok függvényeit, illetve eljárásait metódusoknak nevezzük.