

## Arduino Language – Változók, számok

Egész számok	Méret	Tört számok
<b>byte</b>	8 bits	
<b>int</b>	16 bits	
<b>long</b>	32 bits	<b>float</b>
<b>long long</b>	64 bits	<b>double</b>
	128 bits	<b>long double</b>

Még több info és függvény:

<https://www.arduino.cc/reference/en/>

## Arduino Language – Változók, egyéb

<b>char</b>	8 bit – egy karakter tárolása
<b>string</b>	szöveg tárolása
<b>bool</b>	false vagy true (0 vagy 1)

## Arduino Language – Változók, kiegészítés

<b>const</b>	Ha egy változó elé írjuk, az értéke a program során nem változtatható meg
<b>static</b>	Ha egy lokális változó elé írjuk, csak a függvény első lefutásakor veszi figyelembe
<b>array</b>	Több számot képes eltárolni. 1; 2; 3 dimenziós lehet. pl: int LEDs[4] = {2,3,4,5};

## Arduino Language – Műveletek

Aritmetikai műveletek	
=	egyenlővé tétel
+	összeadás
-	kivonás
/	osztás
*	szorzás
%	osztási maradék pl: 16 % 5 = 1

Összehasonlító műveletek	
==	egyenlőség
<	kisebb
<=	kisebb egyenlő
>	nagyobb
>=	nagyobb egyenlő
!=	nem egyenlő
akkor igaz ha teljesül	

Rövidítések	
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x++	x = x + 1
x--	x = x - 1
teljesen ekvivalens	

&& (logika és)		
x	y	x && y
true	true	true
true	false	false
false	true	false
false	false	false

(logika vagy)		
x	y	x    y
true	true	true
true	false	true
false	true	true
false	false	false

! (logika nem)	
x	!x
true	false
false	true

Kiegészítés	
0	false
1	true

## Arduino Language - Struktúra

void <b>setup()</b> {kód}	A sketch indításakor <b>elsőnek behívott függvény</b> , soros port megnyitására, pin üzemmódok megadására, kezdeti állapotok megadására használhatjuk. <b>Egyszer fut le.</b>
void <b>loop()</b> {kód}	A <b>loop()</b> függvény azt csinálja amit a neve mond. <b>Folyamatosan fut</b> , amíg az Arduino be van kapcsolva
<b>if(feltétel)</b> {kód}	Az <b>if()</b> függvény esetén a kód csak akkor fut le, <b>ha a feltétel teljesül</b> . Akár több feltétel megadása is lehetséges (lásd: logikai és/vagy)
<b>else</b> {kód}	Csak <b>if()</b> függvény után, ha az <b>if()</b> <b>feltétele nem teljesül</b> , a program az else függvény kódjába fog belépni
<b>for</b> (inicializálás; <b>feltétel</b> ; <b>változás</b> ) {kód}	<b>inicializálás</b> : ez történik először, és egyszer pl: int x = 0; <b>feltétel</b> : minden alkalommal amikor a loop végére jut, leellenőrzi hogy a feltétel teljesül-e, ha igen akkor a kód lefut és a <b>változás</b> a loop végén megtörténik. Ez egészen addig fut, amíg a feltétel teljesül. pl. for( <b>int x = 0</b> ; <b>x &lt; 10</b> ; <b>x++</b> ){kód} → A kód 10x fut végig

<b>while</b> (feltétel) {kód}	A kód addig fut, <b>amíg a feltétel teljesül</b> , hasznos lehet akkor, amikor szenzorokkal dolgozunk.
<b>switch</b> (változó) { <b>case</b> 1: kód <b>break</b> ; <b>case</b> 2: kód <b>break</b> ;}	<p>A <b>switch()</b> függvény segítségével elérhetjük hogy <b>különböző feltételek között, különböző kód</b> fusson le.</p> <p>Ha létezik egy olyan <b>case</b> állítás, amelyre teljesül hogy értéke megegyezik a <b>változó</b> értékével, akkor az a kód fut le. A <b>break</b> kilép a <b>switch()</b> függvényből.</p> <p>Tehát ha a <b>változó = 2</b>, akkor a <b>case 2:</b> utáni kód fut le.</p>

## Arduino Language - Függvények

Függvény	Változó 1 (var1)	Változó 2 (var2)	Példa	Leírás
<b>#define</b> var1 var2	konstans neve	konstans értéke	<b>#define</b> LedPin 2	Ha valahova LedPin-t írsz a program azt 2-nek fogja venni. Csak a program legelején.
<b>pinMode</b> (var1,var2)	pin száma (0-13)	INPUT (0) OUTPUT (1)	<b>pinMode</b> (2,OUTPUT);	Beállítja a 2-es pin-t kimeneti pinnek
<b>digitalWrite</b> (var1,var2 )	pin száma (0-13)	LOW (0) HIGH (1)	<b>digitalWrite</b> (2,LOW); <b>digitalWrite</b> (2,HIGH);	2-es pin-en nem folyik áram 2-es pin-en fog folyni áram
<b>delay</b> (var1)	idő ms-ben		<b>delay</b> (1000);	1000 ms várakozás (1 s)
<b>random</b> (var1,var2)	minimum	maximum	<b>random</b> (10,100);	10 és 100 közötti random számot ad vissza
<b>digitalRead</b> (var1)	Digital pin száma (0-13)		<b>digitalRead</b> (2);	0 – ha nem folyik áram a 2-es pin-en 1 – ha a 2-es pin-en folyik áram
<b>analogRead</b> (var1)	Analog pin száma (A0 – A5)		<b>analogRead</b> (2);	A 2-es pin-re eső feszültséget adja vissza 0 -1023 között, ahol 0 = 0V 1023 = 5V. Arányosan
<b>analogWrite</b> (var1,var 2)	PWM pin száma (0-13)	0 – 255 között 0 – 5V arányosan	<b>analogWrite</b> (10,128 );	A 10-es pin-re 2.5 V feszültség esik. Kiszámolható: $5/255 * var2$
<b>pulseIn</b> (var1,var2)	pin száma (0-13)	Várt érték LOW (0) HIGH (1)	<b>pulseIn</b> (5,LOW); <b>pulseIn</b> (5,HIGH);	Addig vár amíg az 5-ös pin LOW-ra változik Addig vár amíg az 5-ös pin HIGH-ra változik
<b>Serial.begin</b> (var1)	Sebesség (Baud rate)		<b>Serial.begin</b> (9600);	Soros port megnyitása 9600-as baud rate-el.
<b>Serial.print</b> (var1)	String Szám formátumú változó		<b>Serial.print</b> ("Hello"); <b>Serial.print</b> (var1);	Kiírja a soros monitorra a Hello feliratot. Fontos az idézőjellel! Kiírja soros monitorra var1 változó értékét
<b>Serial.println</b> (var1)	Bármilyen változó		<b>Serial.println</b> (var1);	Kiírja a var1 értékét, utána új sort kezd
<b>Serial.read</b> ()			<b>Serial.read</b> ();	A soros porton érkező karaktert adja vissza
<b>millis</b> ()			<b>millis</b> ();	az indítás óta eltelt időt adja vissza milliszekundumban

