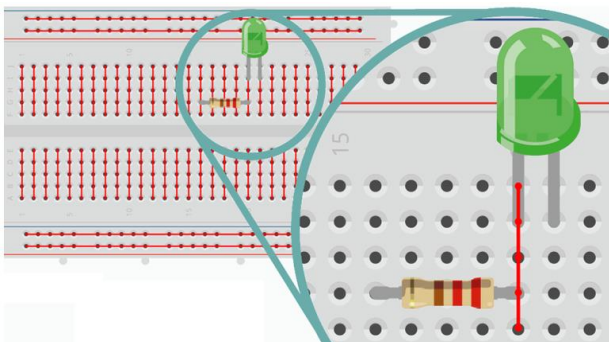
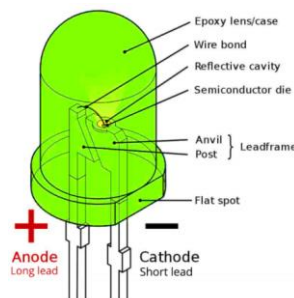


	Funkce	Popis	Příklad
Digital I/O	pinMode	Nastavení pinu jako INPUT, OUTPUT nebo INPUT_PULLUP	pinMode(button_pin, INPUT);
	digitalWrite	Zápis 1 (HIGH) nebo 0 (LOW) na pin	digitalWrite(led_pin, HIGH);
	digitalRead	Čtení stavu digitálního pinu	stav=digitalRead(button_pin);
Analog I/O	analogWrite	Nastavení PWM signálu na pinu	analogWrite (PWMpin, hod); $0 \leq \text{hod} \leq 255$
	analogRead	Čtení stavu analogového pinu (ADC) Vrací hodnotu 0-1023	int stav=analogRead(A0);
Čas	millis	Počet milisekund od startu programu (přeteče po 50 dnech)	cas=millis();
	micros	Počet mikrosekund od startu programu (přeteče po 70min)	cas=micros();
	delay	Pozastavení na daný počet milisekund	delay(1000);
	delayMicroseconds	Pozastavení na daný počet mikrosekund	delayMicroseconds(500);
	pulseIn	Měření délky pulzu na pinu	duration = pulseIn(pin, HIGH);
Sériová linka	Serial.begin	Inicializace a nastavení rychlosti sériového portu	Serial.begin(9600);
	Serial.available	Vrátí počet přijatých bajtů	pocet=Serial.available();
	Serial.print	Odeslání textu nebo dat přes sériovou linku	Serial.print("Ahoj"); nebo Serial.print(cislo);
	Serial.println	Podobně jako Serial.print a navíc se pošle i znak nového řádku	Serial.println("Ahoj"); nebo Serial.println(cislo);
Servo	#include <Servo.h>	Přidání knihovny pro servo	#include <Servo.h>
	Servo servo1	Vytvoření objektu z knihovny Servo	Servo servo1;
	servo.attach	Inicializace serva, nastavení ovládacího pinu	servo1.attach(servo_pin);
	servo.write	Nastavení polohy serva ve stupních	servo1.write(90); //úhel 90 stupnu
LCD displej	#include <Wire.h> #include <LiquidCrystal_I2C.h>	Přidání knihovny pro I2C komunikaci a knihovny pro LCD displej	#include <Wire.h> #include <LiquidCrystal_I2C.h>
	LiquidCrystal_I2C lcd	Vytvoření objektu z knihovny LiquidCrystal_I2C	LiquidCrystal_I2C lcd(0x27, 16, 2);
	lcd.begin	Inicializace displeje	lcd.begin();
	lcd.clear	Vymazání displeje	lcd.clear();
	lcd.setCursor	Nastavení pozice kurzoru (znak, řádek)	lcd.setCursor(0,1);
	lcd.print	Vypsání textu nebo čísel na displeji	lcd.print(„Ahoj!“);

## Nepájivé pole (breadboard)



## Polarita LED diody



Datové typy	
boolean	logická hodnota (true/false)
byte	8-bitová hodnota (0÷255)
int	celé číslo (-32768÷32767)
long	Dlouhé celé číslo ( $-2^{31} \div 2^{31}$ )
unsigned long	Dlouhé celé číslo ( $0 \div 2^{32}$ )
float	reálné číslo ( $-3,4 \cdot 10^{38} \div 3,4 \cdot 10^{38}$ )
char	znak
String	řetězec znaků
Proměnné	
int age = 25;	
float temp = 36.5;	
char znak = 'A';	
Pole	
<pre>int myNumbers[] = {25, 50, 75, 100}; myNumbers[0] = 12; for (int i=0; i&lt;4; i++){   Serial.println(myNumbers[i]); }</pre>	
Další syntaxe	
// jednořádkový komentář	
/* víceřádkový komentář */	
#define LED_PIN 13	
Aritmetické operátory	
+	součet
-	rozdíl
*	součin
/	podíl
%	modulo - zbytek po dělení
=	přiřazení
++	inkrementace
--	dekrementace
+=	součet a přiřazení (a+=b je jako a = a+b)
-=	rozdíl a přiřazení (a-=b je jako a = a-b)
*=	součin a přiřazení (a*=b je jako a = a*b)
/=	rozdíl a přiřazení (a/=b je jako a = a/b)
Porovnání	
>	větší
<	menší
>=	větší nebo rovno
<=	menší nebo rovno
==	rovná se
Logické operátory	
&&	a (AND)
	nebo (OR)
Netisknutelné znaky	
\n	nový řádek
\r	návrat na začátek řádku
\t	odsazení (tabulátor)

Příklad Arduino programu	
<pre>int led_pin = 13; // definice proměnné s číslem pinu  // funkce setup() se spustí jen jednou, po resetu void setup() {   pinMode(led_pin, OUTPUT); // nastaví pin jako výstup }  // funkce loop() se bude volat stále dokola void loop() {   digitalWrite(led_pin, HIGH); // nastav na pinu HIGH-&gt;5V   delay(1000); // počkej 1000ms   digitalWrite(led_pin, LOW); // nastav na pinu LOW-&gt;0V   delay(1000); // počkej 1000ms }</pre>	
Řídící struktury (podmínky, cykly)	
if	<p>Neúplná podmínka</p> <pre>if (podmínka) { ... }</pre> <p>if(a==b)</p> <pre>{   Serial.println("Obě čísla jsou stejná"); }</pre>
if-else	<p>Úplná podmínka</p> <pre>if (podmínka) { ... } else { ... }</pre> <p>if(a==b)</p> <pre>{   Serial.println("Obě čísla jsou stejná"); } else {   Serial.println("Čísla se navzájem liší."); }</pre>
for	<p>Cyklus se známým počtem opakování</p> <pre>for(start; podmínka; krok) { ... }</pre> <pre>for(int i=0; i&lt;=10; i++) {   Serial.println(i); }</pre>
while	<p>Cyklus s neznámým počtem opakování a podmínkou na začátku</p> <pre>while(podmínka) { ... } int i=1; while(i&lt;=10) {   Serial.println(i);   i++; }</pre>
switch	<p>Přepínač podle hodnoty proměnné</p> <pre>switch (klavesa) {   case 1: Serial.println(„Klávesa jedna.“);     break;   case 2: Serial.println(„Klávesa dvě.“);     break;   case 8: Serial.println(„Klávesa tři.“);     break;   default: Serial.println("Neplatné číslo."); }</pre>
break	V těle cyklu for a while umožňuje přeskočit zbytek cyklu a celý cyklus ukončit. Program pak pokračuje dalšími příkazy za ukončeným cyklem.
continue	V těle cyklu umožňuje přeskočit zbytek těla cyklu a pokračovat od začátku cyklu