

1. Tipovi podataka

TIP	ŠIRINA (U BITIMA)	OPSEG VREDNOSTI
char	8	-128..127 ($-2^7..2^7-1$)
unsigned char (isto što i byte)	8	0..255 ($0..2^8-1$)
int	16	-32768..32767 ($-2^{15}..2^{15}-1$)
unsigned int	16	0..65536 ($0..2^{16}-1$)
long	32	-2147483648..2147483647 ($-2^{31}..2^{31}-1$)
unsigned long	32	0..4294967296 ($0..2^{32}-1$)
float (isto što i double)	32	realni brojevi
boolean	8	true, false

2. Promenljive i nizovi

Definicija promenljive:

tip naziv = poč.vrednost; //inicijalizacija je opciona

char c = '@';

int a, b, c = 5;

float trica = 6.75;

boolean logicka_promenljiva;

- Promenljiva koja je definisana u okviru neke funkcije je **lokalna**. Moguće joj je pristupiti samo iz funkcije u kojoj je definisana i vrednost joj se gubi po izlasku iz funkcije.
- Promenljiva koja je definisana izvan (tj. iznad) svih funkcija je **globalna**. Moguće joj je pristupiti iz svih funkcija i statičkog je karaktera, što znači da zadržava vrednost i po izlasku iz funkcije gde joj je dodeljena vrednost.

Definicija niza:

tip naziv[br.članova] = poč.vrednosti; //inicijalizacija je opciona

char ime1[10];

char ime2[32] = "Steva";

int fibonaci[8] = {1, 1, 2, 3, 5, 8, 13, 21};

PRIMER: Sabiranje celih brojeva

```
void setup()
{
  Serial.begin(9600); //inicijalizacija serijskog porta
}

void loop()
{
  int a, b;

  //unos prvog sabirka
  Serial.print("Unesite prvi sabirak: ");
  while(Serial.available() == 0); //ceka da korisnik posalje nesto
  delay(100);
  a = Serial.parseInt(); //ocitavanje broja iz serijskog bafera
  Serial.println(a);

  //unos drugog sabirka
  Serial.print("Unesite drugi sabirak: ");
  while(Serial.available() == 0); //ceka da korisnik posalje nesto
  delay(100);
  b = Serial.parseInt(); //ocitavanje broja iz serijskog bafera
  Serial.println(b);

  Serial.print("Zbir = ");
  Serial.println(a + b);
}
```

PRIMER: Ispis prvih 8 članova Fibonačijevog niza

```
int fibonaci[8] = {1, 1, 2, 3, 5, 8, 13, 21};
int i;

void setup()
{
  Serial.begin(9600); //inicijalizacija serijskog porta
}

void loop()
{
  if(i < 8)
  {
    Serial.println(fibonaci[i]);
    i++;
  }
  delay(1000);
}
```

PRIMER: Unos stringa (niza karaktera)

```
void setup()
{
  Serial.begin(9600); //inicijalizacija serijskog porta
}

void loop()
{
  char ime[32];
  int duzina;

  Serial.println("Unesite svoje ime: ");
  while(Serial.available() == 0); //ceka da korisnik posalje nesto
  delay(100);
  duzina = Serial.available();
  Serial.readBytes(ime, duzina); //ocitavanje stringa iz serijskog bafera
  ime[duzina] = 0; //terminacija stringa

  Serial.print("Zdravo, ");
  Serial.print(ime);
  Serial.println("! :)");
}
```

3. Uslovni izrazi: if.else, switch..case

PRIMER: Provera koji igrač je pobedio

```
int prvi = 5, drugi = 3;

void setup()
{
  Serial.begin(9600);

  if (prvi == drugi)
  {
    Serial.println("Nereseno!");
  }
  else if (prvi > drugi)
  {
    Serial.println("Prvi igrac je pobedio!");
  }
  else
  {
    Serial.println("Drugi igrac je pobedio!");
  }
}

void loop()
{
}
```

PRIMER: Ocene u školi

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int ocena;

  Serial.print("Unesite ocenu: ");
  while (Serial.available() == 0);
  delay(100);
  ocena = Serial.parseInt();
  Serial.println(ocena);

  switch (ocena)
  {
    case 1:
      Serial.println("Nedovoljan!");
      break;
    case 2:
      Serial.println("Dovoljan!");
      break;
    case 3:
      Serial.println("Dobar!");
      break;
    case 4:
      Serial.println("Vrlo dobar!");
      break;
    case 5:
      Serial.println("Odlican!");
      break;
    default:
      Serial.println("Uneli ste nepostojecu ocenu!");
  }
}
```

4. Petlje: while, do..while, for

PRIMER: String + petlje

```
void setup()
{
  Serial.begin(9600); //inicijalizacija serijskog porta
}

void loop()
{
  char ime[32];
  int duzina, index;

  Serial.println("Unesite svoje ime: ");
  while(Serial.available() == 0); //ceka da korisnik posalje nesto
  delay(100);
  duzina = Serial.available();
  Serial.readBytes(ime, duzina); //ocitavanje stringa iz serijskog bafera
  ime[duzina] = 0; //terminacija stringa

  index = 0;
  Serial.println("Spreda: ");
  while(ime[index] != 0)
  {
    Serial.println(ime[index]);
    index++;
  }

  Serial.println("otpozadi: ");
  do
  {
    index--;
    Serial.println(ime[index]);
  }
  while(index != 0);

  Serial.print("Zdravo, ");
  for(index = 0 ; index < duzina; index++)
    Serial.print(ime[index]);
  Serial.println("! :)");

  Serial.print("Vozdra, ");
  for(index = duzina-1 ; index >=0; index--)
    Serial.print(ime[index]);
  Serial.println("! :P");
}
```

PRIMER: ASCII tabela (videti primer za Serial.println() u skripti)

5. Funkcije

Definicija funkcije:

```
povratni_tip naziv_funkcije(lista_parametara )
```

```
{  
  
    //telo funkcije  
  
    return povratna_vrednost;  
  
}
```

- Lista parametara je opciona, tj. parametri mogu, ali i ne moraju da postoje
- Ukoliko funkcija ne vraća nikakvu vrednost, povratni tip je **void**. Primeri funkcija bez povratnih vrednosti i bez parametara su *void setup()* i *void loop()*.

PRIMER: Pitagorina teorema

```
float a = 3.0, b = 4.0;  
float c;  
  
float pitagora(float kateta1, float kateta2)  
{  
    float hipotenuza = sqrt(kateta1 * kateta1 + kateta2 * kateta2);  
    return hipotenuza;  
}  
  
void setup()  
{  
    Serial.begin(9600); //inicijalizacija serijskog porta  
  
    Serial.print("Duzina prve katete je ");  
    Serial.println(a);  
  
    Serial.print("Duzina druge katete je ");  
    Serial.println(b);  
  
    Serial.print("Duzina hipotenuze je ");  
    Serial.println(pitagora(a, b));  
}  
  
void loop()  
{  
}
```