

## Vežbe 12

1. Napraviti objekat iterator tako da on podržava sledeće ponašanje:

```
const it = makeRangeIterator(1, 10, 2);
let result = it.next();
while (!result.done) {
    console.log(result.value); // 1 3 5 7 9 (11)
    result = it.next();
}
```

2. Kreirati generator koji generiše prirodne brojeve.

3. Napraviti funkcije

- a. *raditest()* koja nakon 3 sekunde na konzolu ispisuje "test završen"
- b. *pregledajTest()* koja nakon 2 sekunde na konzolu ispisuje "test pregledan"
- c. uskladiti funkcije tako da druga započinje nakon prve upotrebom povratnog poziva
- d. uskladiti funkcije tako da druga započinje nakon prve upotrebom *Promise* objekata

4. Analizirati ponašanje koda kada se menjaju vrednosti parametara *dostupno* i *uspesnoPlacanje*.

```
function proveruDostupnostProizvoda() {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            const dostupno = true; // Recimo da je proizvod dostupan
            if (dostupno) {
                resolve("Proizvod je dostupan");
            } else {
                reject("Proizvod nije dostupan");
            }
        }, 2000);
    });
}
```

```
function procesirajPlacanje() {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            const uspesnoPlacanje = true; // Recimo da je plaćanje uspešno
            if (uspesnoPlacanje) {
                resolve("Placanje uspešno");
            } else {
                reject("Placanje nije uspelo");
            }
        });
    });
}
```

```

    }, 3000);
  });
}

function izvrsiDostavu() {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve("Dostava izvršena");
    }, 5000);
  });
}

proveriDostupnostProizvoda()
  .then((poruka) => {
    console.log(poruka); // "Proizvod je dostupan"
    return procesirajPlacanje();
  })
  .then((poruka) => {
    console.log(poruka); // "Placanje uspešno"
    return izvrsiDostavu();
  })
  .then((poruka) => {
    console.log(poruka); // "Dostava izvršena"
  })
  .catch((error) => {
    console.error("Došlo je do greške: ", error);
  });
};

```

##### 5. Date su funkcije:

```

function dobaviSliku() {
  return new Promise( (prihvati, odbij) => {
    setTimeout ( () => prihvati("Uspesno dobavljena slika"), 2000)
  })
}

function dobaviUsername() {
  return new Promise( (prihvati, odbij) => {
    setTimeout ( () => prihvati("Uspesno dobavljen username"), 500)
  })
}

function dobaviPrijatelje() {
  return new Promise( (prihvati, odbij) => {
    setTimeout ( () => prihvati("Uspesno dobavljeni prijatelji"), 3000)
  })
}

```

Napisati program koji

- a) poziva ove tri funkcije i ispisuje na konzolu razrešenu vrednost one *Promise* varijable koja se prva razreši
- b) poziva ove tri funkcije i ispisuje na konzolu razrešene vrednost svih *Promise* objekata nakon što se sve funkcije zavrse

## Rešenja

1.

```
function makeRangeIterator(start, end, step) {  
  let trenutniBrojac = start;  
  let kraj = false;  
  
  let r = {  
    next: function () {  
      trenutniBrojac += step;  
      if (trenutniBrojac > end) {  
        kraj = true;  
      }  
      let result = {  
        done: kraj,  
        value: trenutniBrojac  
      }  
      return result;  
    }  
  }  
  return r;  
}
```

2.

```
function* generatorPrirodnihBrojeva() {  
  let i = 1;  
  while (true) {  
    yield i;  
    i++;  
  }  
}
```

```
let g = generatorPrirodnihBrojeva();
console.log(g.next());
```

3.

a)

```
function raditest() {
  setTimeout( () => console.log("test završen"), 3000 );
}
```

b)

```
function pregledajTest() {
  setTimeout( () => console.log("test pregledan"), 2000 );
}
```

c)

```
function raditest(callback) {
  setTimeout( () => {
    console.log("test završen");
    callback();
  }, 3000 );
}

function pregledajTest() {
  setTimeout( () => console.log("test pregledan"), 2000 );
}
```

```
raditest(pregledajTest)
```

d)

```
function raditest() {
  return new Promise( (razresi, odbij) => {
    setTimeout( () => {
      console.log("test završen");
      razresi();
    }, 3000 );
  });
}
```

```
function pregledajTest() {
```

```
return new Promise ( (razresi, odbij) => {  
  setTimeout ( () => {  
    console.log("test pregledan");  
    razresi();  
  }, 2000)  
} )  
}
```

```
raditest().then(() => pregledajTest());
```

5.

```
let p1 = dobaviSliku();  
let p2 = dobaviUsername();  
let p3 = dobaviPrijatelje();
```

- a) `Promise.race( [p1, p2, p3]).then ( (vr) => console.log(vr));`
- b) `Promise.all( [p1, p2, p3]).then ( (vr) => console.log(vr));`