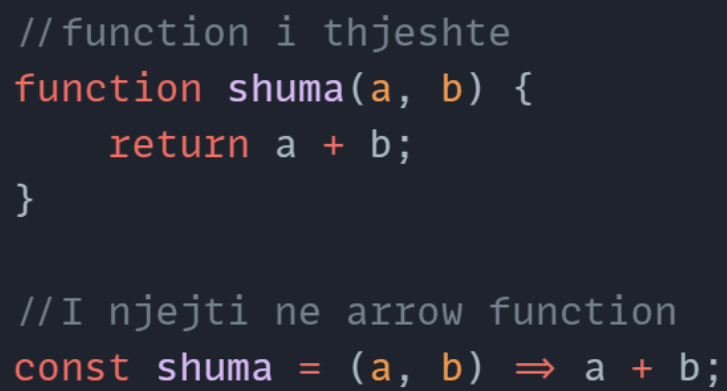


## 1. Funkcioni me parametrin emri i cili kthen nje mesazh pershendetes

```
//funksion në JavaScript të quajtur pershendetje,  
//i cili pranon një parametër emri dhe kthen një mesazh përshëndetës.  
  
function pershendetje(emri) {  
    return `Përshëndetje, ${emri}!`;  
}  
console.log(pershendetje("Arber"));
```

## 2. Konvertimi nga funksioni i thjeshte ne funksion "arrow function"



```
//function i thjeshte  
function shuma(a, b) {  
  return a + b;  
}  
  
//I njejti ne arrow function  
const shuma = (a, b) => a + b;
```

3. Funkcion me parametric “emri” por qe ka nje vlere te parazgjedhur “Mik”

```
//një funksion pershendet, i cili merr një parametër emri,  
//por ka një vlerë të parazgjedhur "Mik" nëse nuk i jepet ndonjë argument.  
  
function pershendet(emri = "Mik") {  
    return `Përsëndetje, ${emri}!`;  
}
```

snappify.com

4. Shembull i funksionit me vlere dhe me callback qe shumezon me numrin 2

```
//një funksion punoMeTeDhena,  
//i cili pranon një numër dhe një funksion callback që shumëzon numrin me 2.  
  
function punoMeTeDhena(numri, callback) {  
    return callback(numri);  
}  
  
function dyfisho(numri) {  
    return numri * 2;  
}  
  
console.log(punoMeTeDhena(5, dyfisho)); // 10
```

snappify.com

5. Konvertimi i nje callback ne nje Promise qe kthen “te dhenat u moren” pas 1 sekondi:

```
//Callback i cili shfaq tekstin "te dhenat u moren" pas 1 sekondi

function merrTeDhena(callback) {
  setTimeout(() => {
    callback("Të dhënat u morën");
  }, 1000);
}

// funksionin e mësipërm duke përdorur Promise:
function merrTeDhena() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve("Të dhënat u morën");
    }, 1000);
  });
}

merrTeDhena().then(console.log);
```

6. Shembulli i nje funksioni qe perdor “async” dhe kthen pergjegjen e specifikuar pas 2 sekondash:

```
//një funksion merrMesazh që përdor async
//dhe pas 2 sekondash kthen "Përshëndetje nga funksioni asinkron".

async function merrMesazh() {
  return new Promise(resolve => {
    setTimeout(() => {
      resolve("Përshëndetje nga funksioni asinkron");
    }, 2000);
  });
}

merrMesazh().then(console.log);
```

7. Shembull i funksionit asinkron qe ben Fetch te dhenat nga API dhe e kthen emrin e perdoruesit me console.log:

```
//një funksion merrPerdoruesin, i cili merr të dhëna nga
//https://jsonplaceholder.typicode.com/users/1 dhe
//ben fetch emrin e përdoruesit në console.log.

async function merrPerdoruesin() {
  let response = await fetch("https://jsonplaceholder.typicode.com/users/1");
  let data = await response.json();
  console.log(data.name);
}

merrPerdoruesin();
```

8. Shembulli i nje funksioni asinkron dhe si mund te perdoret metoda try...catch... per te kapur gabimet (error handling)

```
//funksion asinkron
async function merrTeDhena() {
  let response = await fetch("https://jsonplaceholder.typicode.com/posts/1");
  let data = await response.json();
  console.log(data);
}

//funksionin e mësipërm që përdorë try...catch për të kapur gabimet (error handling):
async function merrTeDhena() {
  try {
    let response = await fetch("https://jsonplaceholder.typicode.com/posts/1");
    let data = await response.json();
    console.log(data);
  } catch (error) {
    console.error("Gabim:", error);
  }
}
```

9. Shembulli i konvertimit të funksionit të thjeshtë për kapjen e të dhënave nga API - në një funksion asinkron:

```
//funksion i thjeshtë për kapjen e të dhënave përmes API
function fetchPost() {
    fetch("https://jsonplaceholder.typicode.com/posts/1")
        .then(response => response.json())
        .then(data => console.log(data))
        .catch(error => console.error("Error:", error));
}

//Konvertimi funksionin e mësipërm që të përdorë async/await:
async function merrPostime() {
    try {
        let response = await fetch("https://jsonplaceholder.typicode.com/posts/1");
        let data = await response.json();
        console.log(data);
    } catch (error) {
        console.error("Gabim:", error);
    }
}

merrPostime();
```

snappify.com

### Dallimi mes Funksionit/Kodit Sinkron dhe atij Asinkron:

**Kodi sinkron** ekzekutohet rresht pas rreshti, duke pritur që secila komandë të përfundojë përpara se të kalojë te tjetra.

**Kodi asinkron** lejon që disa procese të ndodhin në sfond pa ndaluar ekzekutimin e pjesës tjetër të programit. Për shembull, `setTimeout`, `fetch`, dhe `async/await` janë teknika asinkrone që lejojnë që programi të vazhdojë ndërkohë që një operacion i gjatë përfundon.