

```
Async function() {  
  Await ...  
}
```

# სინქრონიზაცია და ასინქრონიზაცია

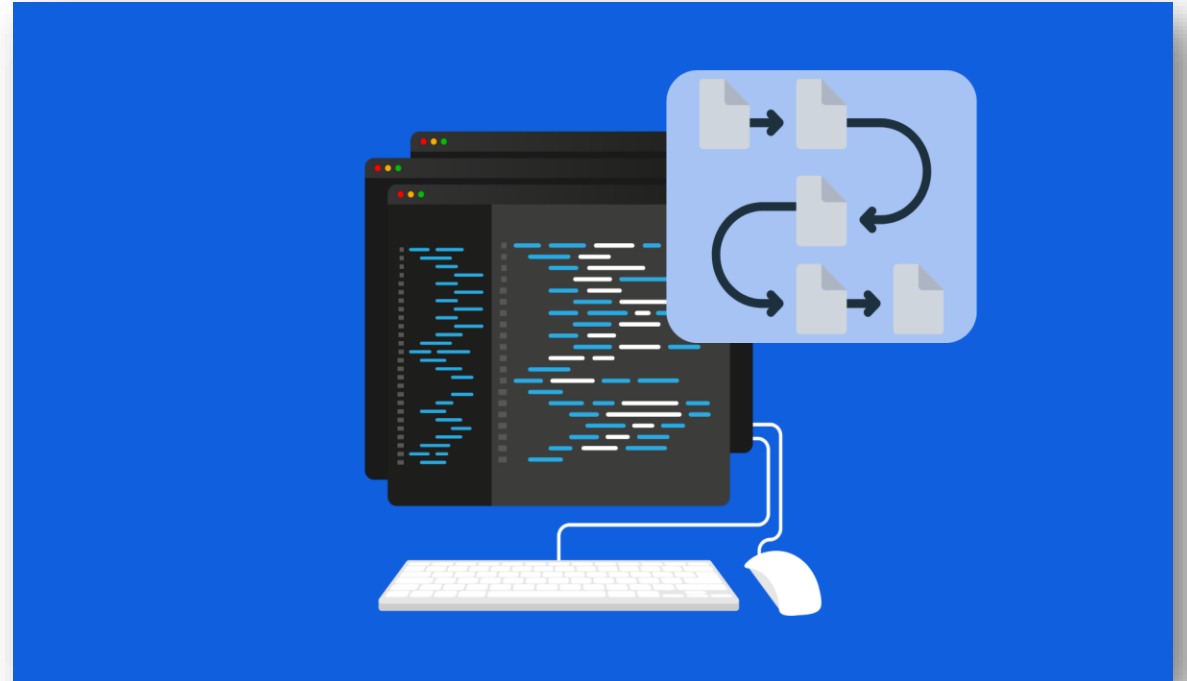


```
query  
"SELECT * FROM marks WHERE subject = ?"  
function (datasetsWithSubject) {  
  if (datasetsWithSubject.length > 0) {  
    subjectAverage = 0;  
    datasetsWithSubjectLength = datasetsWithSubject.length;  
    datasetsWithSubject.forEach((dataset) => {  
      subjectAverage += parseFloat(dataset.mark);  
    });  
  }  
}
```

პრეზენტაცია მოამზადა: ნიკოლოზი გორეზიანი

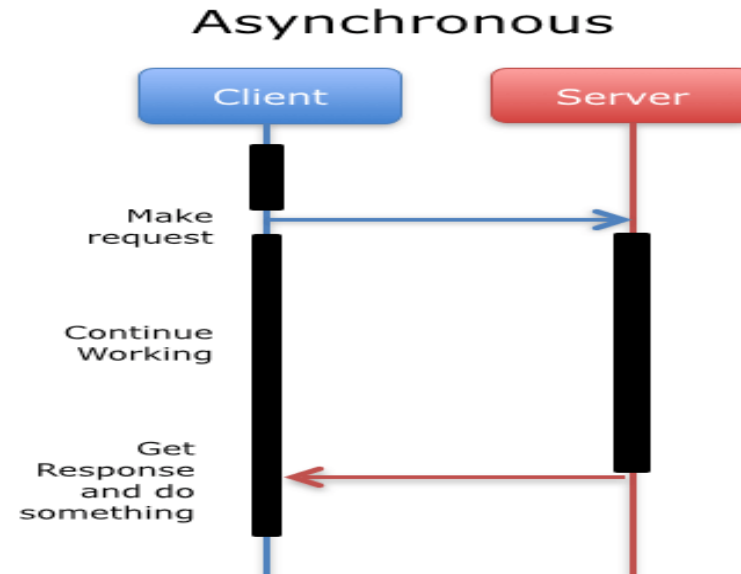
# რა არის სინქრონიზაცია?

- სინქრონიზაცია არის პროცესი, რომლის დროსაც სხვადასხვა ოპერაცია სრულდება თანმიმდევრობით. თითოეული ოპერაცია უნდა დასრულდეს, სანამ დაიწყება შემდეგი.



# რა არის ასინქრონიზაცია?

- ასინქრონიზაცია ნიშნავს, რომ ოპერაციები არ არის დამოკიდებული ერთიმეორის დასრულებაზე, ანუ ერთი ოპერაცია შეიძლება დასრულდეს იმავე დროს, როცა სხვა ოპერაცია მიმდინარეობს.



# ცხოვრების მაგალითი - სინქრონიზაცია

- სამზარეულოში მუშაობა
  - თუ აკეთებთ საჭმელს, ჯერ უნდა მოამზადოთ ინგრედიენტები, შემდეგ ჩართოთ ღუმელი, გამოაცხოთ და მხოლოდ ამის შემდეგ შეგიძლიათ მიირთვათ.
  - თითოეული ქმედება დამოკიდებულია წინა პროცესის დასრულებაზე.



# ცხოვრების მაგალითი - ასინქრონიზაცია

- თეფშების გარეცხვა ჭურჭლის სარეცხ მანქანაში
  - თეფშებს ათავსებთ მანქანაში და პროცესი იწყება. ამავედროულად, შეგიძლიათ აკეთოთ სხვა საქმეები (მაგალითად, უყუროთ ფილმს) მანამ, სანამ მანქანა მუშაობს.



# სინქრონიზაცია კოდში

```
1 function syncTask() {  
2   console.log("Task 1");  
3   console.log("Task 2");  
4 }  
5 syncTask();
```

შედეგი:

- Task 1
- Task 2

თითოეული მოქმედება  
სრულდება  
თანმიმდევრობით.

# ასინქრონიზაცია კოდში

```
1 console.log("Task 1");
2
3 setTimeout(() => {
4   console.log("Task 2");
5 }, 1000);
6
7 console.log("Task 3");
```

შედეგი:

- Task 1
- Task 3
- Task 2 (1 წამში)

Task 2 იწყება ასინქრონულად და ამ დროის განმავლობაში Task 3 უკვე სრულდება.

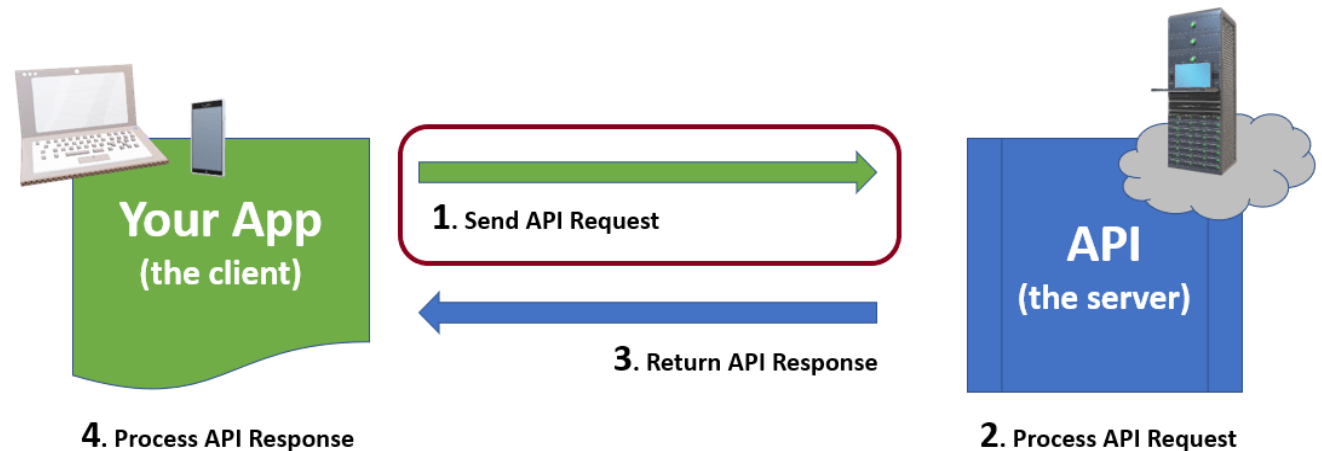
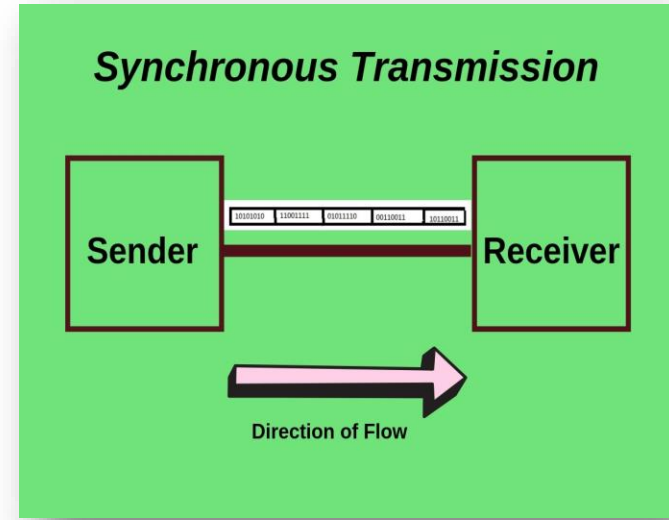
# გამოყენების შემთხვევები

სინქრონიზაციის გამოყენების შემთხვევები:

- მონაცემთა დამუშავება (ფაილების წაკითხვა/ჩაწერა)
- ერთ ეტაპიანი პროცესები, სადაც ყოველი ნაბიჯი არის დამოკიდებული წინა ეტაპის დასრულებაზე.

ასინქრონიზაციის გამოყენების შემთხვევები:

- ვებ აპლიკაციებში სერვერიდან მონაცემთა მიღება
- რთული პროცესები, რომლებსაც შეუძლიათ დამოუკიდებლად მუშაობა (მაგ. API გამოძახებები).





# მადლობა ყურადღებისთვის!

```
selected");e.selecting=false;e.selected=true;e.startselected=true;e.trigger("selected",  
false));a.extend(a.ui.selectable,{version:"1.8.16"}))(jQuery);  
(function(a){a.widget("ui.sortable",a.ui.mouse,{widgetEventPrefix:"sort",options:  
{appendTo:"parent",axis:false,connectWith:false,containment:false,helper:"default",items:">  
*",opacity:false,placeholder:false,revert:false,scroll:true,scrollSensitivity:20,scrollSpeed:20,  
{var d=this.options;this.containerCache={};this.element.addClass("ui-sortable");  
this.refresh();this.floating=this.items.length?d.axis=="x"||/left|right/.test(this.items[0].item.css("display")):false;this.offset=this.element.offset();this._mouseInit();  
cell/.test(this.items[0].item.css("display")):false;this.offset=this.element.offset();this._mouseInit();  
sortable-disabled").removeData("sortable").unbind(".sortable");this._mouseDestroy();  
item");return this},_setOption:function(d,c){if(d=="disabled"){this.options[d]=c;this.widget().removeClass("ui-sortable-disabled");  
"disabled"){this.options[d]=c;this.widget().addClass("ui-sortable-disabled");  
a.widget.prototype._setOption.apply(this,arguments)},_mouseCapture:function(d,c){if(this.refresh()&&this._mouseMoveCapture(d,c)){return true;this._refreshPositions();this._mouseStart(d,c);  
false;if(this.options.disabled||this.options.type=="static")return false;this._refreshPositions();this._mouseStart(d,c);  
(if(a.data(this,"sortable-item")==h){e=a(this);return true;this._refreshPositions();this._mouseStart(d,c);  
e){var g=false;a(this._removeCurrentsFromItems());return true;this._refreshPositions();this._mouseStart(d,c);  
this.currentContainer=this;this.refreshPositions();this.helper.css("position","absolute");this._mouseStart(d,c);  
this.offset=this.helper.offset();this.helper.css("position","absolute");this._mouseStart(d,c);  
relative:this._getRelativeOffset();this.helper.css("position","absolute");this._mouseStart(d,c);  
mark,cursor);this.helper.css("position","absolute");this._mouseStart(d,c);  
mark,cursor);this.helper.css("position","absolute");this._mouseStart(d,c);
```