

# 24.01.29

🕒 작성 일시	@2024년 1월 29일 오전 10:05
🏷 태그	

## 함수 정의하기

10+20 = 30

undefined+undefined = NaN

```
<title>Document</title>
<script>
  //함수
  //1.익명 함수

  //2.function
  let test = function (){
    document.write("<h2>함수 정의하기</h2>")
  }

  function add(num1, num2){
    document.write("<p>${num1}+${num2} = ${num1 + num2}<p>`");
  }
  test();
  add(10, 20);
  add();
</script>
</head>
```

반지름이 10인 원의 넓이: 314  
사각형의 넓이 : 200  
5000  
반지름이 10인 원의 넓이: 314  
undefined

```
<title>Document</title>
<script>
  //함수의 매개변수와 반환값
  function circle10(){
    document.write(`반지름이 10인 원의 넓이: ${10*10*3.14} <br>`);
  }

  circle10();

  let rectangle = function(width, height){
    return width * height;
  }

  document.write("사각형의 넓이 : " + rectangle(10, 20) + "<br>");

  function multiply( x, y=5, z=10){
    console.log("z : " + z);
    return x*y*z;
  }

  document.write(multiply(10,50) + "<br>")

  document.write(circle10())
</script>
```

```

<script>
  //화살표 함수
  // let func = (매개변수) => {함수본문}
  let circle10 = () =>{
    | document.write(`반지름이 10인 원의 넓이 : ${10* 10* 3.14}<br>`);
  }

  circle10();
  let circle00 = function(radius){
    | return radius * radius * 3.14;
  }

  let circle0 = radius => radius * radius * 3.14;
  document.write (circle0(10) + "<br>");

  let rectangle = (width, height) => width * height;
  let triangle = (width, height = 5) => width * height / 2;

  document.write(triangle(14, 6))
</script>

```

```

반지름이 10인 원의 넓이 : 314
314
42

```

```

<script>
  //1. 함수 방식
  const student = {
    이름: "만식이",
    학년: 3,
    toString: function(){
      return "<p>"
        + "이름 : " + this.이름 + "<br>"
        + "학년 : " + this.학년 + "<br>"
        + "</p>";
    }
  };

  //2. 익명 함수 방식
  const Member = function(name, id, address){
    this.name = name;
    this.id = id;
    this.address = address;
    this.toString = function(){
      return "<p>"
        + "이름 : " + this.name + "<br>"
        + "아이디 : " + this.id + "<br>"
        + "주소 : " + this.address + "<br>"
        + "</p>";
    }
  };

  //3. 생성자 함수 방식

  function Human(name, age, gender){
    this.name = name;
    this.age = age;
    this.gender = gender;
    this.toString = function(){
      return "<p>"
        + this.name + "(" + this.gender + "-" + this.age + ")"
        + "</P>";
    }
  };

  console.log(student);
  document.write(student);

  const member = new Member("왕씨", "midas", "관악구청");
  let human = new Human("김씨", 48, "남성");

  document.write(member);
  document.write(human);

```

이름 : 만식이  
학년 : 3

이름 : 왕씨  
아이디 : midas  
주소 : 관악구청

김씨(남성-48)

```

<script>
  //생성자는 객체를 초기화 하는 역할을 한다.
  class Member{
    constructor(name, id, address){
      this.name = name;
      this.id = id;
      this.address = address;
    }

    toString(){
      return`
        <p>
          이름: ${this.name}<br>
          아이디: ${this.id}<br>
          주소: ${this.address}
        </p>
      `;
    }
  };

  const Student = class{
    constructor(name, grade){
      this.name = name;
      this.grade = grade;
    }
    toString(){
      return`
        <p>
          이름 : ${this.name}<br>
          학년 : ${this.grade}
        </p>
      `;
    }
  }

  const member = new Member("최가", "lime", "광명");
  const s1 = new Student("임꺽", "3학년");
  const s2 = new Student("허준", "1학년");

  document.write(member);
  document.write(s1);
  document.write(s2);
</script>

```

이름: 최가  
아이디: lime  
주소: 광명

이름 : 임꺽  
학년 : 3학년

이름 : 허준  
학년 : 1학년

```

</script>
<script>
  let timer
  let clockStart = function(){

    timer = window.setInterval(function(){
      let date = new Date();
      document.getElementById("result").innerHTML = date.toLocaleTimeString();
    }, 1000)
  }

  //onload를 사용하여 body를 먼저 읽고 실행하게 해준다.
  window.onload = function(){
    document.querySelector("#result").innerHTML
    ="3초 후에 시계가 자동으로 시작됩니다.";

    let timeOut =window.setTimeout(function(){
      clockStart();
    }, 3000)
  }
</script>

```

타이머 정지  
오후 1:37:22

```

<script>
  window.onload = function(){
    let plan = document.getElementById("plan")
    console.log(plan.children);

    const liTags = document.getElementsByTagName("li");
    for(let i=0; i< liTags.length; i++){
      console.log(liTags[i].innerHTML);
    }

    const doneList = document.getElementsByClassName("plan-list");
    console.log(doneList);

    const content = document.querySelector(".done-list");
    console.log(content);

    const contentAll = document.querySelectorAll(".done-list");
    console.log(contentAll);

    contentAll.forEach((value, i) =>{
      console.log(i, value);
    })

  };

```

### DOM Tree

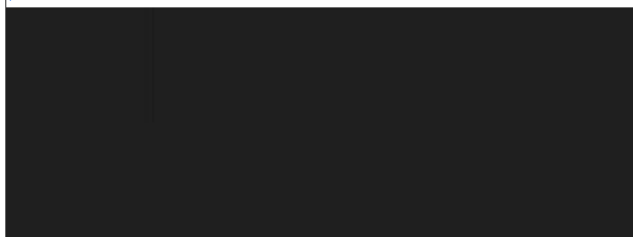
프로그래밍 스터디 계획

- JavaScript 함수
- JavaScript DOM
- JavaScript Event
- jQuery 함수 선택자
- jQuery Effect

프로그래밍 스터디 완료

- HTML 기본 태그
- HTML5 시맨틱 태그
- CSS3 선택자
- CSS3 박스모델
- JavaScript 제어문

HTMLCollection(5) [li.plan-list, li.plan-list, li.plan-list, li.plan-list, li.plan-list]	javascript:08_01Templa
JavaScript 함수	javascript:08_01Templa
JavaScript DOM	javascript:08_01Templa
JavaScript Event	javascript:08_01Templa
jQuery 함수 선택자	javascript:08_01Templa
jQuery Effect	javascript:08_01Templa
HTML 기본 태그	javascript:08_01Templa
HTML5 시맨틱 태그	javascript:08_01Templa
CSS3 선택자	javascript:08_01Templa
CSS3 박스모델	javascript:08_01Templa
JavaScript 제어문	javascript:08_01Templa
HTMLCollection(5) [li.plan-list, li.plan-list, li.plan-list, li.plan-list, li.plan-list]	javascript:08_01Templa
▶ <li class="done-list"> ☰ </li>	javascript:08_01Templa
NodeList(5) [li.done-list, li.done-list, li.done-list, li.done-list, li.done-list]	javascript:08_01Templa
0 ▶ <li class="done-list"> ☰ </li>	javascript:08_01Templa
1 ▶ <li class="done-list"> ☰ </li>	javascript:08_01Templa
2 ▶ <li class="done-list"> ☰ </li>	javascript:08_01Templa
3 ▶ <li class="done-list"> ☰ </li>	javascript:08_01Templa
4 ▶ <li class="done-list"> ☰ </li>	javascript:08_01Templa



```

</script>
<script>
  window.onload = function(){
    const plan = document.querySelector('#plan');
    const li1 = document.createElement("li");
    li1.textContent = "오라클 sql 스터디 준비";
    plan.appendChild(li1); // 뒤에 추가

    const li2 = li1.cloneNode(true);
    plan.prepend(li2); // 가장 앞에 추가
    li2.textContent = "JavaScript 제어문 복습";

    const li3 = li2.cloneNode(true);
    li3.textContent = "함수 만들기 집중 공략";
    const planChild1 = plan.children[1];
    planChild1.before(li3); // plan 첫 자식 요소 다음

    plan.firstChild.remove(); // 첫번째 자식요소 삭제

    const done = document.querySelector("#done");
    done.prepend(plan.firstChild); //앞에 추가
    done.children[1].before(plan.firstChild); //1번째 다음에 추가
  }
</script>

```

## 프로그래밍 스터디 계획

- JavaScript DOM
- JavaScript Event
- jQuery 함수 선택자
- jQuery Effect
- 오라클 sql 스터디 준비

## 프로그래밍 스터디 완료

- 함수 만들기 집중 공략
- JavaScript 함수
- HTML 기본 태그
- HTML5 시멘틱 태그
- CSS3 선택자
- CSS3 박스모델
- JavaScript 제어문

데이터변경