

2021 ThreeJS 001

- Set-up part 1 // new project, node install

!Tips n Links

- ※ learn ThreeJS, WebGL, typescript // ready ~> VSCode, Git, NodeJS, typescript
- ※ typescript: `npm install -g typescript` / 확인: `git --version, node -v, npm -v, tsc -v`(오류 시 `tsc.cmd -v`)
- ※ ThreeJS <https://threejs.org/> ※ sbcode <https://sbcode.net/threejs/create-threejs-project/>
- ※ 유데미 강좌 <https://www.udemy.com/course/threejs-tutorials/> ※ 다른 강의 <https://threejs-journey.com/>

Typescript

javascript+IntelliSense features (인텔리센스, 지능형코드완성, 기능 집합)
will use TS => to generate JS suitable for webpack to bundle codes into a single file
TS 새로고침: F1, TypeScript: Restart TS server

ThreeJS

- ※ 프로젝트 만들기:
node> three
@type/three
네이티브 WebGL을 최적화하고 컨트롤을 제공하는 라이브러리 // three.min.js 연결
`mkdir 플젝폴더 > cd 폴더 > dir /ls > ① npm init` (package.json 생성) > code .
② `npm install three --save-dev` => json에 devDependencies에 추가
③ `mkdirs dist > client + server` ④ `dist> client`에 index.html 만들기¹⁾
⑤ `mkdirs src > client + server` ⑥ `src> client`에 client.ts²⁾ + tsconfig.json³⁾ 만들기
⑦ `npm install @types/three --save-dev`

<https://sbcode.net/threejs/create-threejs-project/>
<https://sbcode.net/threejs/add-initial-scripts/>

<< 코드 복사

1)./dist/client/index.html 2)./src/client/tsconfig.json 3)./src/client/client.ts

1)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport"
      content="width=device-width,
      initial-scale=1" />
    <title>Three.js TypeScript Tutorials</title>
    <style>
      body {
        overflow: hidden;
        margin: 0px;
      }
    </style>
  </head>
  <body>
    <script type="module" src="bundle.js"></script>
  </body>
</html>
```

2)

```
{
  "compilerOptions": {
    "moduleResolution": "node", //how to find 라이브러리... node다~
    "strict": true //instantiate variable..? 추천사항
  },
  "include": ["**/*.ts"] <- TS compiler가 여기에 파일 넣을거=>client.ts
}
```

3)

```
import * as THREE from 'three'
const scene = new THREE.Scene()
const camera = new THREE.PerspectiveCamera(
  75,
  window.innerWidth / window.innerHeight,
  0.1,
  1000
)
camera.position.z = 2
const renderer = new THREE.WebGLRenderer()
renderer.setSize(window.innerWidth, window.innerHeight)
document.body.appendChild(renderer.domElement)
const geometry = new THREE.BoxGeometry()
const material = new THREE.MeshBasicMaterial({
  color: 0x00ff00,
  wireframe: true,
})
const cube = new THREE.Mesh(geometry, material)
scene.add(cube)
window.addEventListener('resize', onWindowResize, false)
function onWindowResize() {
  camera.aspect = window.innerWidth / window.innerHeight
  camera.updateProjectionMatrix()
  renderer.setSize(window.innerWidth, window.innerHeight)
  render()
}
function animate() {
  requestAnimationFrame(animate)
  cube.rotation.x += 0.01
  cube.rotation.y += 0.01
  render()
}
function render() {
  renderer.render(scene, camera)
}
animate()
```

/* -----<lang: en>----- */

dist: distribution 배포, 분포 / instantiate설명, 예시 classes, variables / IntelliSense features코드 자동완성?정보?