

Komentarai

```
// Initialize
window.onload(=> {
  // Adding event listeners to element
  let element = document.getElementById("id");
  element.onkeypress(myFunction);

  // Removing special characters from element text
  element.innerText = element.innerText.replace(/^[^a-zA-Z ]/g, '');
})
```




```
window.onload(initialize);
```

```
function removeSpecialCharacters(element){
  element.innerText = element.innerText.replace(/^[^a-zA-Z ]/g, '');
}
```


```
function initialize(){
  let element = document.getElementById("id");
  addEventListeners(element);
  removeSpecialCharacters(element);
}
```

```
function addEventListeners(element){
  element.onkeypress(myFunction);
}
```

Magic Numbers



```
function checkPassword(password){  
  if(password.length < 7){  
    console.log('OK!');  
  }  
}
```



```
function checkPassword(password){  
  const MAX_PASSWORD_SIZE = 7;  
  
  if(password.length < MAX_PASSWORD_SIZE){  
    console.log('OK!');  
  }  
}
```

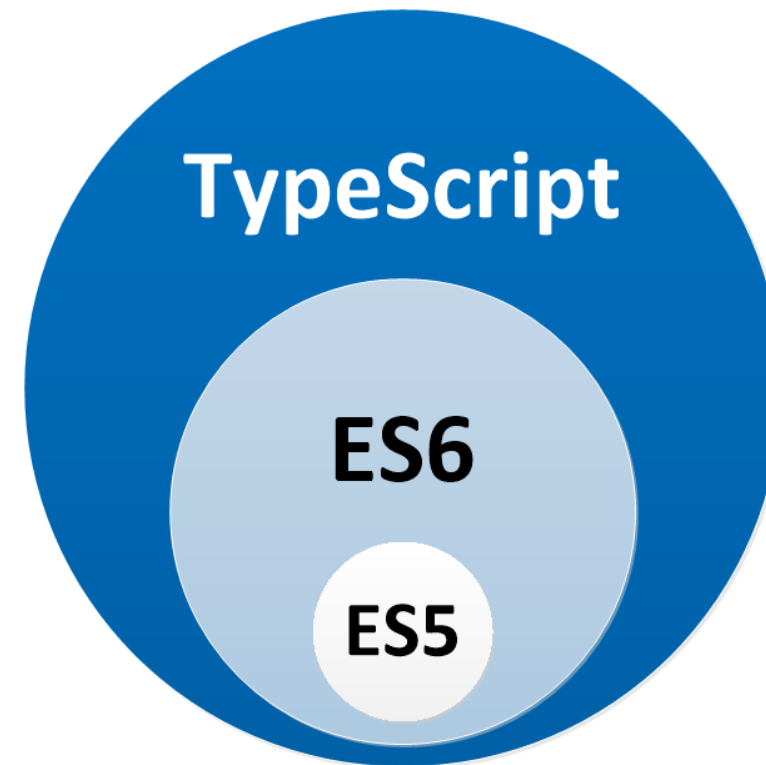
FRONT-END
Academy
by Present Connection

TS



TypeScript

- ◆ Superset of JavaScript
- ◆ 2012 m. Spalio 1 (Microsoft)
- ◆ .ts
- ◆ Open source
- ◆ Kompiluoja į Js



JS Tipai

```
let academy;
```

```
academy = 2018;
```

```
let academy: any  
academy = 'PC';
```



Tipai

```
let number: number  
let number = 7;
```

```
let number : number;
```

```
number = "Academy started";
```


```
[ts] Type '"Academy started"' is not assignable to type 'number'.
```

```
let number: number
```


Tipai

```
function startAcademy(name : string)
{
    console.log(`${name} started !!!`);
}
```


```
startAcademy(7);
```



```
startAcademy();
```




```
startAcademy("Present connection");
```




Optional parameter

```
function startAcademy(name?: string)
{
    console.log(`${name} started !!!`);
}
```


```
startAcademy(7);
```



```
startAcademy();
```




```
startAcademy("Present connection");
```




Tipai

```
function getCurrentYear() : number  
{  
    let date = new Date();  
    return date.getFullYear();  
}
```

```
let result : string;  
result = getCurrentYear();
```



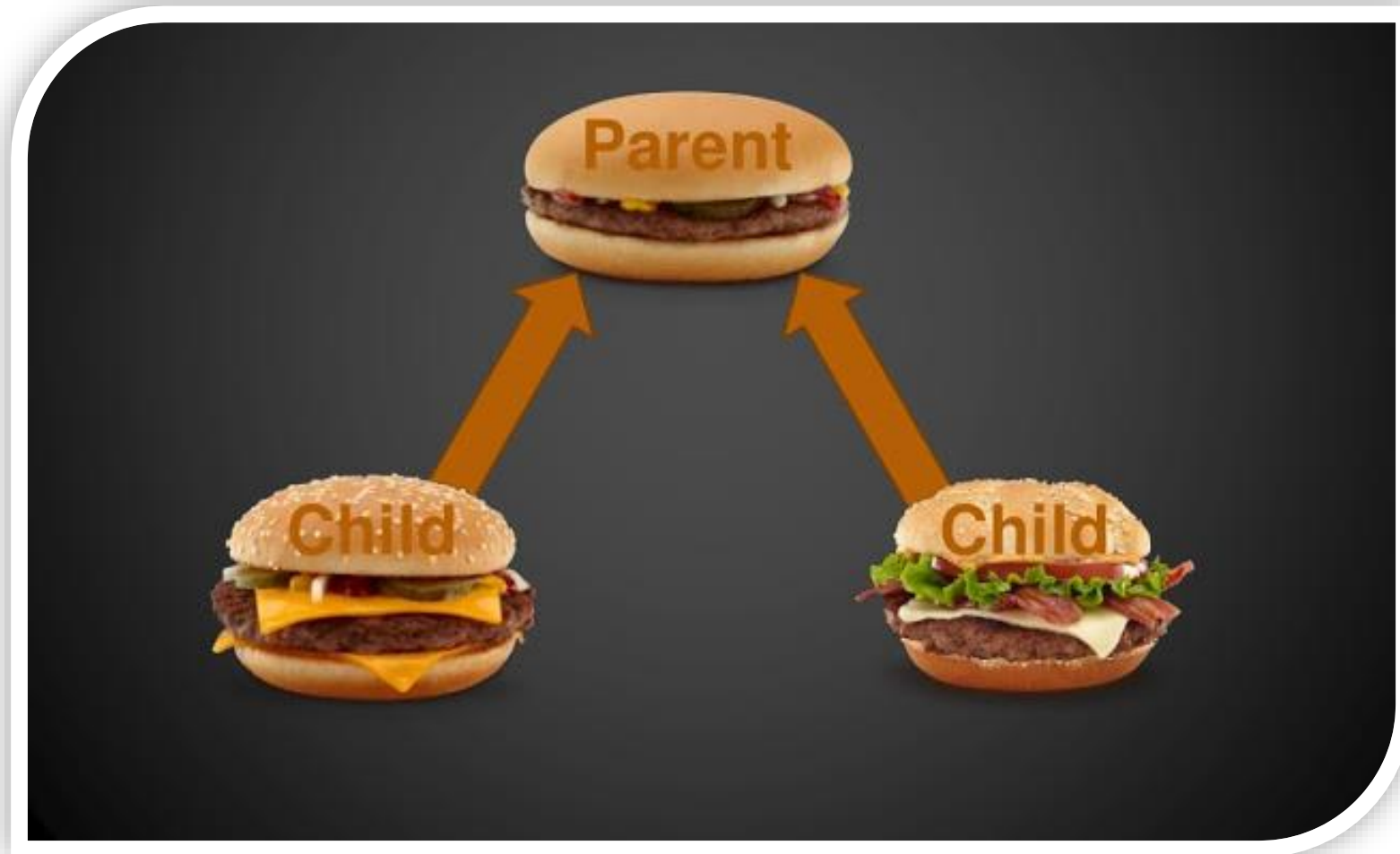
```
let result : number;  
result = getCurrentYear();
```



Tipai

Tipas	Aprašymas	Pavyzdys
Array	Reikšmių masyvas	let list: number[] = [1, 2, 3];
Tuple	Masyvas, kurio pirmų elementų tipai yra apibrėžti, o tolimesni turi atitikti vieną iš tipų.	let x: [string, number] = ["7", 7, 7, 8, "8"];
Enum	Būdas skaičiams suteikti gražius pavadinimus	enum Color {Red, Green, Blue};
Any	Kai nežinom koks bus tipas	let notSure: any = 4; notSure = false;
Void	Naudojamas funkcijom kurios negrąžina reikšmės	function warnUser(): void {alert()};
Never		function error (message: string): never { throw new Error(message); }
Null/Undefined		let u: undefined = undefined

Abstrakti klasè



Abstrakti klasè

```
abstract class Animal
{
    name: string;
    age: number;

    makeSound()
    {
        console.log("Sound");
    }
}
```

```
class Cat extends Animal
{
    breed: string;

    makeSound()
    {
        console.log("miau");
    }
}
```

Interface

Interface
Human



Implements

Class Man



Implements

Class Woman



Interface

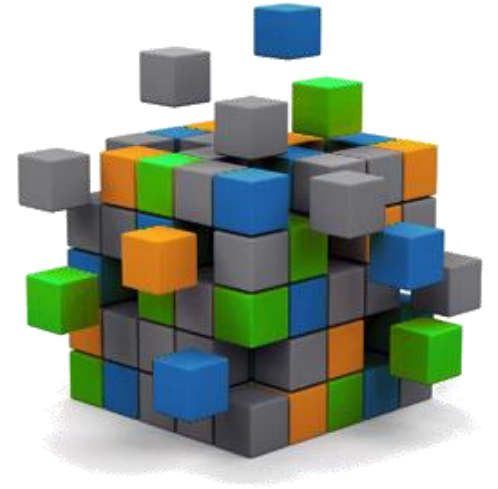
```
interface Animal
{
    name: string;
    age: number;
    makeSound();
}
```

```
class Dog implements Animal
{
    name : string;
    age: number;

    makeSound()
    {
        console.log("bark");
    }
}
```

Moduliai

- ◆ Moduliai yra vykdomi savo lokaliame scope. Tai reiškia kad klasės, kintamieji, funkcijos ir t.t. nėra matomi moduliui išorėje.
- ◆ Norint panaudoti modlius išorėje naudojame export/import.



Moduliai

```
export class Human
{
    name:string;

    move()
    {
        console.log("Move");
    }
}
```

Naudojantis export galima eksportuoti daug modulių iš failo

```
import { Human } from './academy-modules';
```


Moduliai

```
export default class Human
{
    name:string;

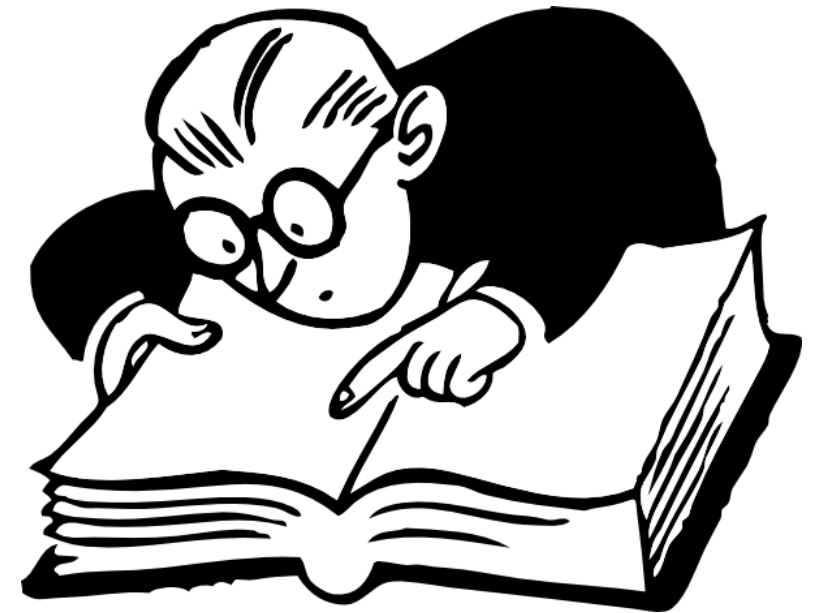
    move()
    {
        console.log("Move");
    }
}
```

Naudojantis export default galima eksportuoti tik vieną modulį iš failo

```
import People from './academy-modules-default';
```

Declaration files

- ◆ Skirti pateikti typescript'ui informaciją apie bibliotekos modulius (pavadinimai, tipai, argumentai ir t.t).
- ◆ Naudojamas tik kompiliatoriaus
- ◆ .d.ts



Konfigurācija

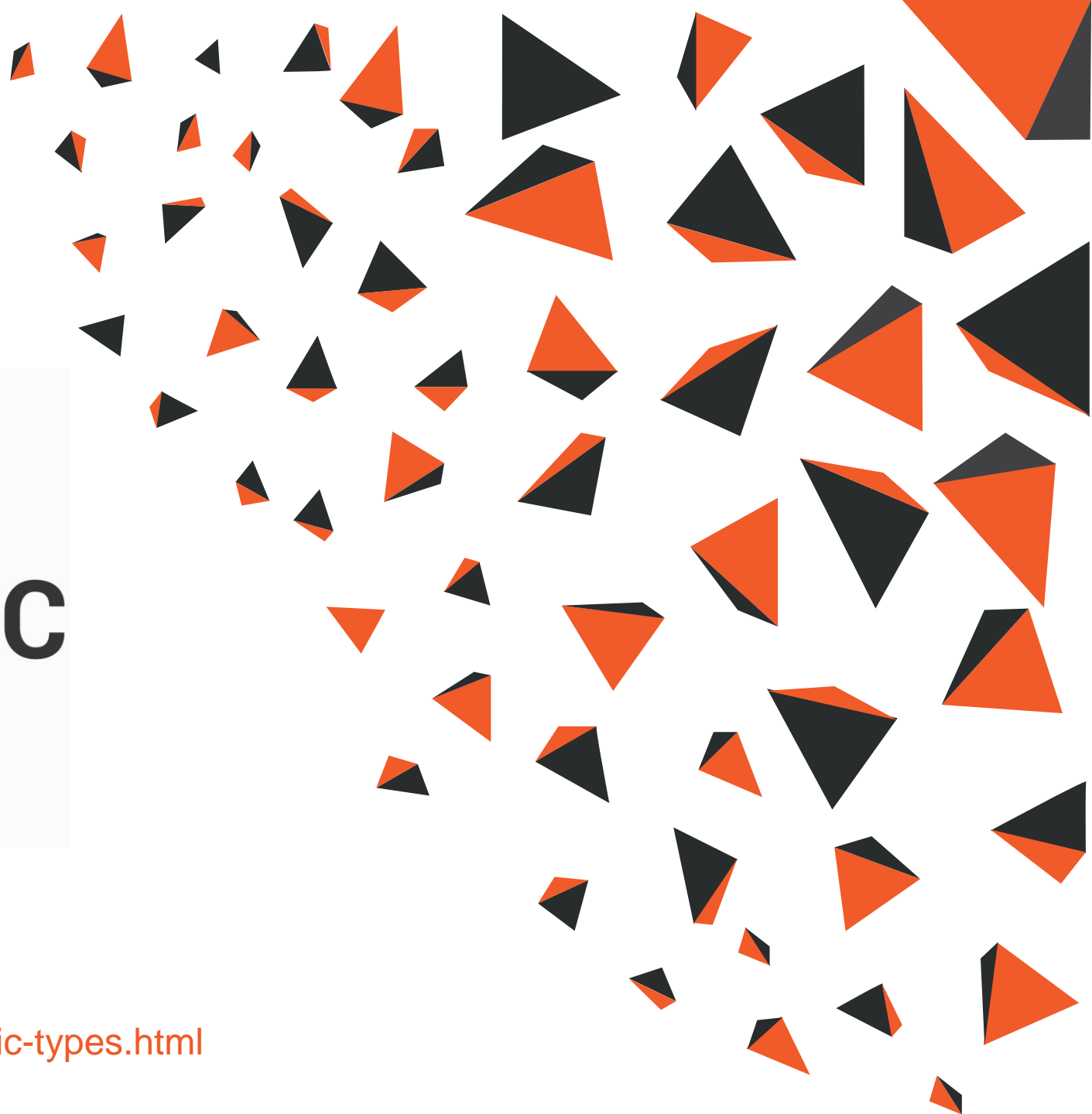
- ◆ npm install typescript
- ◆ Pridēti failā: tsconfig.json
- ◆ Paleisti komandā: tsc

```
"compilerOptions": {  
  "strictNullChecks": true,  
  "noImplicitAny": true,  
  "watch": true,  
  "sourceMap": true,  
  "target": "es5",  
  "outDir": "./js",  
  "allowJs": true,  
  "module": "es6"  
},  
"compileOnSave": true,  
"include": [  
  "./scripts/*"  
]  
}
```

DEMO

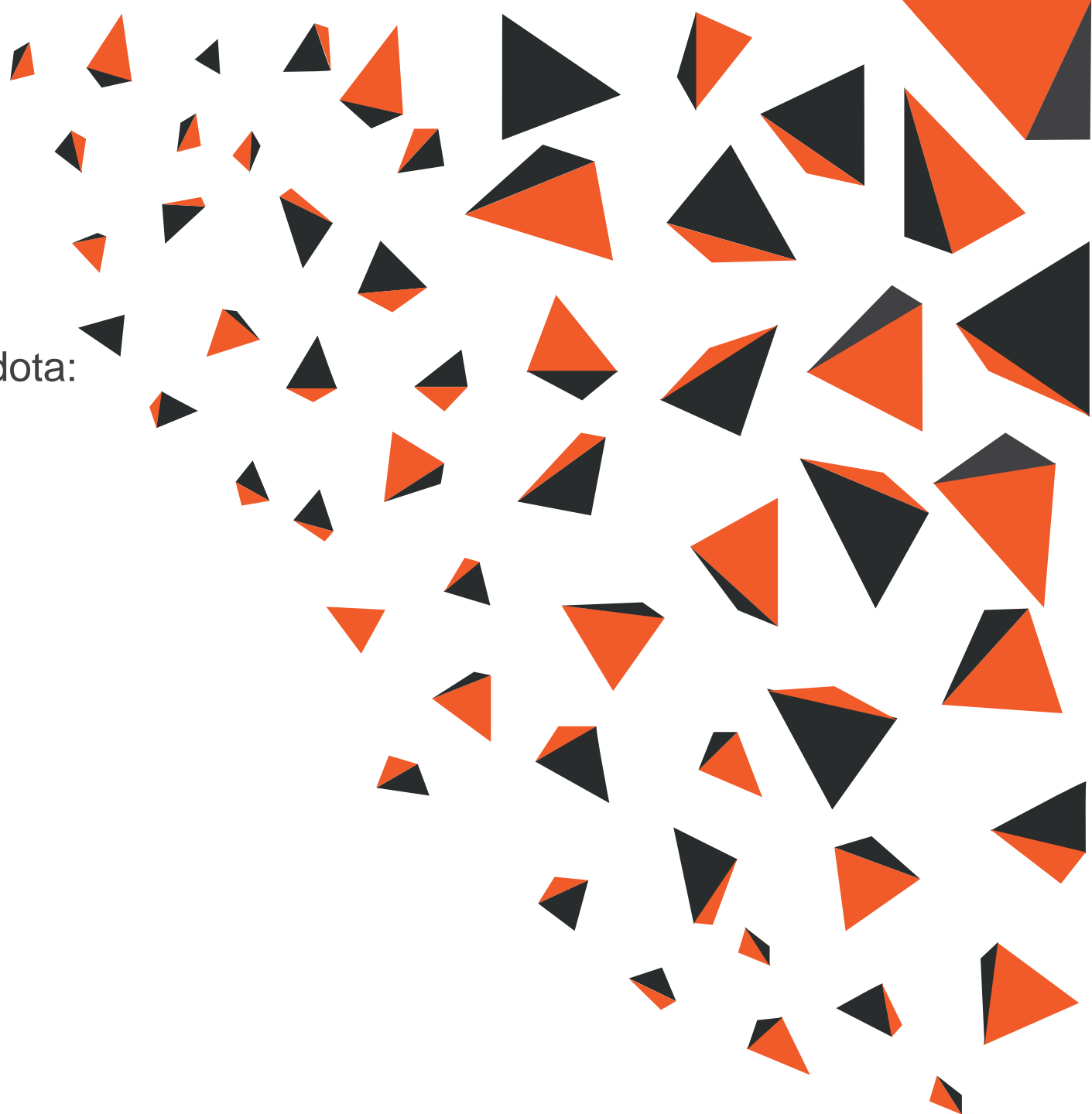
TYPESCRIPT: SETUP & BASIC USAGE

<https://www.typescriptlang.org/docs/handbook/basic-types.html>



Workshop

- ◆ Gausite legacy code failą
- ◆ Perrašyti kodą į TypeScript. Turi būti panaudota:
 - ◆ Types
 - ◆ Moduliai
 - ◆ Class
 - ◆ Ecma6...



Namų darbai

- ◆ Klientas nori įgyvendinti paskolų skaičiuoklę (kiek reikės mokėti kas mėnesį).
- ◆ Paskolų tipai: greitoji paskola, būsto paskola, vartojimo paskola.
- ◆ 3 skirtingi html puslapiai kiekvienai paskolai. Kiekviename po formą ją užpildžius paskaičiuojame kiek reikės mokėti ir pateikiama kita info.
- ◆ Reikalavimai:
 - ◆ GP: Formos laukai: paskolos dydis ir terminas. Palūkanos 20%. Neturi viršyti 5000 eurų ne daugiau kaip 2 metams.
 - ◆ BP: Formos laukai: paskolos dydis, terminas, alga, kiek vaikų. Palūkanos 2%.Makisimalus teriminas 30 metų. Maksimalus paskolos dydis paskaičiuojamas pagal algą.
 - ◆ VP: Formos laukai: paskolos dydis,terminas,daikto tipas (dropdown) kuriam imama. Neturi viršyti 10000 eurų ne daugiau kaip 5 metams. Priklausoma nuo daikto tipo skiriasi palūkanos.
 - ◆ Būtinai panaudoti abstract class arba interface (arba abu) ir kiekvienai paskolai po klasę.