

Taakgroep: Javascript Advanced

Taak: De game opdelen in meerdere modules

De game bestaat nu nog uit één module. In de loop van de ontwikkeling van de game leidt dit tot een erg grote module. Daarom delen we de game op in modules, en belangrijk: alleen de hoofdmodule `Game` blijft beschikbaar als variabele in de global scope.

Aanpak

Je voegt aan de bestaande module `Game` een submodule toe door de module een property te geven, die ook weer een module is.

- Geef de module `Game` een property `Reversi` met als waarde een module:

```
Game.Reversi = (function(){
    console.log('hallo, vanuit module Reversi')
})();
```

- Geef de module `Game.Reversi` een private variabele `configMap`.
- Geef de module `Game.Reversi` een publieke `init` function die bij aanroep de private function `privateInit` uitvoert.
- Test in de console of `Game.Reversi.init` beschikbaar is.

Ondersteunende informatie

Toelichting op module pattern

Een uitstekende en uitgebreide toelichting op het module pattern: mastering de module pattern.

Private function in module

Een private function is alleen in de scope van de module beschikbaar:

```
const Game = (function(){

    const privateInit = function(){
        // doing private things...
    }

})();
```

Function publiek vanuit module

Door een waarde te retourneren vanuit de module kan een functie beschikbaar worden gemaakt voor de outer scope:

```
const Game = (function(){  
  
    const privateInit = function(){ ... }  
  
    return {  
        publicInit : privateInit  
    }  
  
})();
```

Parameters van een module

Waarden kunnen als parameters aan een module worden verstrekt

```
const apiUrl = 'url/super/duper/game'  
  
const Game = (function(url){ //2. waarde worde als parameter aan functie meegegeven  
  
    let configMap = {  
        api : url, //3. parameter url wordt bewaard in een object  
    }  
  
    ...  
  
})(apiUrl) //1. waarde wordt aan de module meegegeven
```