

# Transformation Priority Premises

From Nothing ...



to Mutation



{ } → nil

no code at all

```
public String wrap(String s, int length) {  
    return null;  
}
```

code that employs nil

```
public String wrap(String s, int length){  
    return "";  
}
```

**nil → constant**

**nil**

```
public String wrap(String s) {  
    return "";  
}
```

**constant**

```
public String wrap(String s){  
    return "at least something";  
}
```

**constant → constant<sup>+</sup>**

**constant**

```
public Food giveMeFood(Money money){
```

```
    return }
```



**constant<sup>+</sup>**

```
public Food giveMeFood(Money money){
```

```
    return
```



```
 +
```



```
}
```

**constant → scalar**

**constant**

```
public Food giveMeFood(Money money){
```

```
    return 
```

```
}
```

**scalar**

```
public Food giveMeFood(  ){
```

```
    return 
```

```
}
```

# statements → statements

## statement

```
public Food giveMeFood(   ){
```

```
    return   
}
```

## statements

```
public Food giveMeFood(  ,  ,  ){
```

```
    return max (    )  
}
```

# unconditional → conditional

## unconditional

```
public Drink giveMeADrink(){  
    return   
}
```

## conditional

```
public Drink giveMeADrink( ){  
    if ( TGIF ){  
        return   
    }  
    return   
}
```

# scalar → array

scalar

```
public Food giveMeFood(){
```

```
    return  +   
}
```

array

```
public Food giveMeFood(){
```

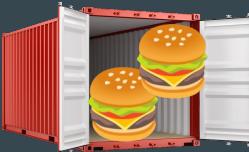
```
    return [  ,  ]  
}
```

# array → container

array

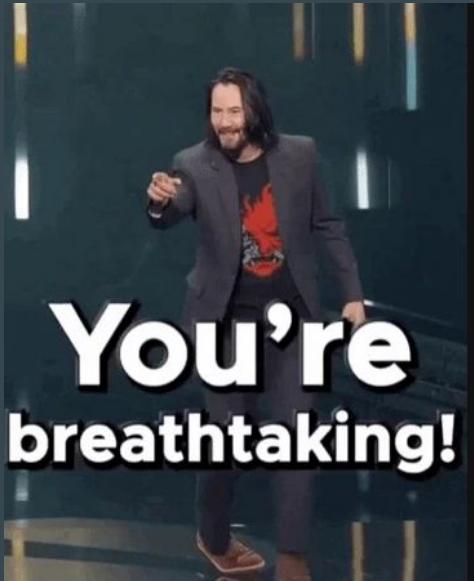
```
public Food[] giveMeFood(){  
    return [   ]  
}
```

container

```
public FoodContainer giveMeFood(){  
    return   
}
```

**statement → tail-recursion**

**statement**



**tail-recursion**



**statement** → loop

**statement**

**loop**

if (! Saturday && ! Sunday) {

(for Day day:workday){

    return work

}

    return work

}

**statement → recursion**

**statement**

**recursion**

**if (! Saturday && ! Sunday) {**

**return work**

**}**



# expression → function

expression

putDoughInForm()

putFormInOven()

waitForAnHour()

function

```
bakeACake(){  
    putDoughInForm()  
    putFormInOven()  
    waitForAnHour()  
}
```

**variable** → mutation

**variable**

`var animal =`



**mutation**

`animal =`

