

# Teaser

Functional Programming 2019/20

Frank Staals



Universiteit Utrecht

[Faculty of Science  
Information and Computing  
Sciences]

# Welcome to **Functional Programming 2019/2020**



**Universiteit Utrecht**

[Faculty of Science  
Information and Computing  
Sciences]

# Specification

$f\ 0 = 1$

$f\ n = n * f(n-1)$



# Specification

`f 0 = 1`

`f n = n * f(n-1)`

`choose n k = f(n) / (f(k)*f(n-k))`



# Specification

`f 0 = 1`

`f n = n * f(n-1)`

`choose n k = f(n) / (f(k)*f(n-k))`

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$



# Implementation

```
int f(int n) {  
    int result = 1;  
    for (int i = n; i > 0; i--)  
        result *= i;  
    return result;  
}
```



# Implementation

```
int f(int n) {  
    int result = 1;  
    for (int i = n; i > 0; i--)  
        result *= i;  
    return result;  
}
```

```
int choose(int n, int k) {  
    return f(n)/(f(k)*f(n-k));  
}
```



# Specification

$f\ 0 = 1$

$f\ n = n * f(n-1)$

$\text{choose } n\ k = f(n) / (f(k) * f(n-k))$





# Implementation in Haskell

```
f 0 = 1
```

```
f n = n * f(n-1)
```

```
choose n k = f(n) / (f(k)*f(n-k))
```



# What is Functional Programming?

- ▶ More a *style* than a *paradigm*
- ▶ What code **is** instead of what code **does**



# What is Functional Programming?

- ▶ More a *style* than a *paradigm*
- ▶ What code **is** instead of what code **does**
- ▶ You can write “functional code” in almost any language



# Implementation in Haskell (again)

```
f 0 = 1
```

```
f n = n * f (n - 1)
```

```
n `choose` k = f n / (f k * f (n-k))
```

