

# CLOSURES

CLOSURES (in deutsch Funktionsabschlüsse) sind Konstruktionen einer inneren und einer äußeren Funktion.

Normalerweise wird die innere Funktion definiert, wenn die äußere Funktion aufgerufen wird.

Bei einer Closure kann die innere Funktion jedoch aufgerufen werden unabhängig von der äußeren, dabei schafft sie sich Ihre eigene Funktions-Umgebung.

Anders ausgedrückt: Die in der Closure definierte Funktion merkt sich die Umgebung, in der sie erzeugt wurde.

```
function essen(gericht) {  
  let x = 0;  
  return function () {    // anonyme Funktion  
    x++;  
    console.log(`ich esse ${x} Portionen ${gericht}.`);  
  }  
}
```

```
let essenAusgabe = essen(`Currywurst mit Pommes`);
```

// hier wird 1mal die Funktion essen() ausgeführt und der return zwischengespeichert.

```
essenAusgabe();  
essenAusgabe();  
essenAusgabe();  
essenAusgabe();  
essenAusgabe();
```

// hier wird 5mal die anonyme Funktion ausgeführt die auf den Closure der Funktion essen() zugreift.

// deshalb wird auf der Console ausgegeben:

```
ich esse 1 Portionen Currywurst mit Pommes.  
ich esse 2 Portionen Currywurst mit Pommes.  
ich esse 3 Portionen Currywurst mit Pommes.  
ich esse 4 Portionen Currywurst mit Pommes.  
ich esse 5 Portionen Currywurst mit Pommes.
```

```
essen(`Pommes mit Majo`());  
essen(`Pommes mit Majo`());  
essen(`Pommes mit Majo`());  
essen(`Pommes mit Majo`());  
essen(`Pommes mit Majo`());
```

// hier wird 5mal die Funktion essen aufgerufen, wobei die anonyme Funktion jedes Mal auf einen unterschiedlichen (neuen) Closure zugreift.  
// deshalb wird auf der Console ausgegeben:

ich esse 1 Portionen Pommes mit Majo.  
ich esse 1 Portionen Pommes mit Majo.  
ich esse 1 Portionen Pommes mit Majo.  
ich esse 1 Portionen Pommes mit Majo.  
ich esse 1 Portionen Pommes mit Majo.

**Questions? You're welcome!**

[holger.zerbe@digitalcareerinstitute.org](mailto:holger.zerbe@digitalcareerinstitute.org)