

For

```
for (Initialisierung; Bedingung; Befehlsfolge){  
    //Anweisungen  
}
```

Do- While

```
do {  
    //Anweisungsblock  
} while (Bedingung)
```

While

```
while (Bedingung){  
    //Anweisungsblock  
}
```

If

```
if (Bedingung){  
    //Anweisungsblock  
} else {  
    //Anweisungsblock  
}
```

Switch

```
switch (Ausdruck){  
    case Wert1:  
        //Programmblock  
        break  
    case Wert2:  
        //Programmblock  
        break  
    //usw.  
    default:  
        //Programmblock  
}
```

Array

```
ar a = new Array()  
a[1] = "Januar"  
a[2] = "Februar"  
//usw.  
//z.b. var a = new Array("", "Januar", "Februar", "März")
```

Funktion

```
function Funktionsname(Parameter1, Parameter2){  
    //Programmblock  
    return Wert //ist optional  
}
```

Funktionsbeispiele

```
//BSP1:  
function tag_ausgeben(s){  
    document.write("<" + s + ">")  
}
```

```
//Test-Aufruf
tag_ausgeben("P")
document.write("Winnie Pooh")
tag_ausgeben("/P")
```

```
//BSP2:
function tag(s){
    return "<"+s+">"
}

//Der Aufruf
document.write(tag("P")+"Winnie Pooh"+tag("/P"))
```

```
//BSP3: Funktionen mit Parameter „überladen“ und auswerten
//ist eigentlich ein Workarounds für Netscape-Bugs ^^
function tag(){
    var returnstring = ""
    for (var i=0; i<tag.arguments.length; i++)
        returnstring += "<"+tag.arguments[i]+">"
    return returnstring
}

//Der Aufruf //Ergebnis ist "<FONT SIZE='2'><FONT FACE='Arial'><B>"
tag("FONT SIZE='2'", "FONT FACE='Arial'", "B")
```

Globale Variable auch als (andere !!!) lokale Variable verwenden

```
var a=0 //globale Variable
var b=0 //globale Variable
function test(){
    a++ //Zugriff auf globale Variable a
    var b=1 //Neue lokale Variable
        //hat mit der globalen Variablen b nichts zu tun da nur lokal
    var c=0 //Neue lokale Variable
}
```