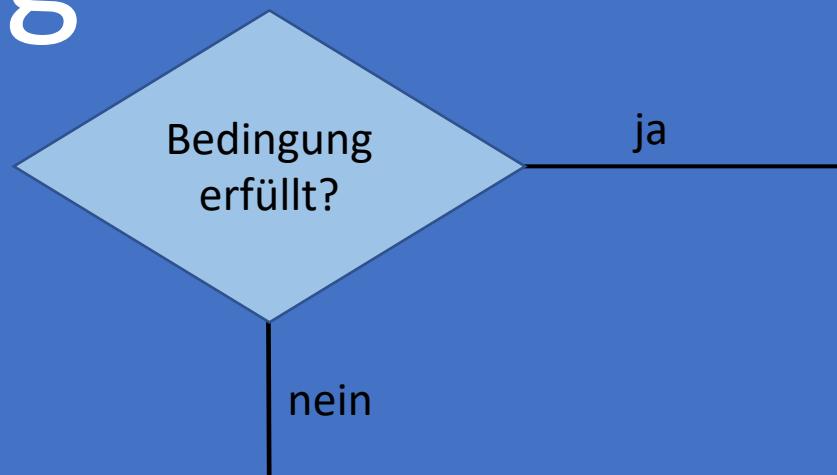
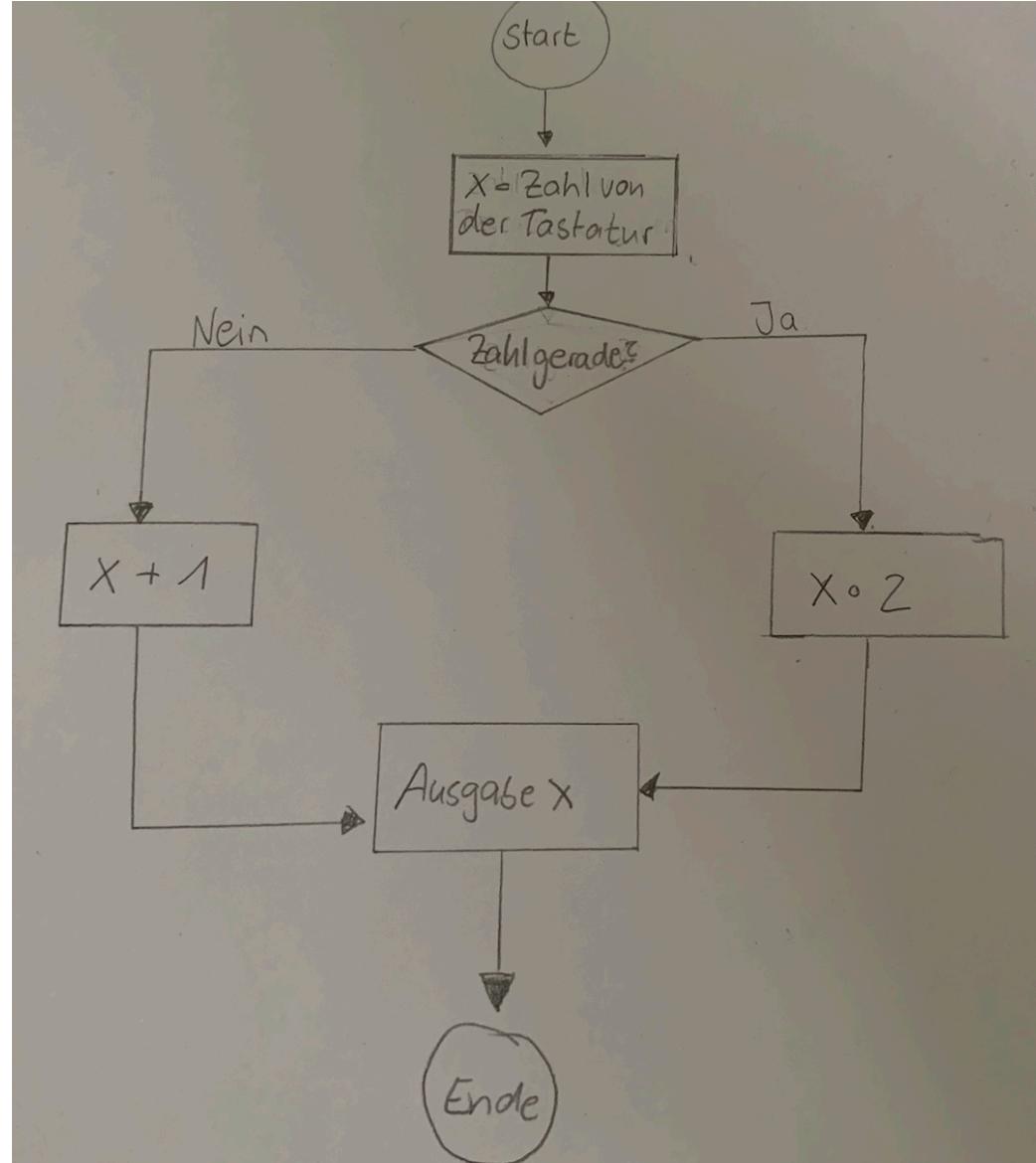


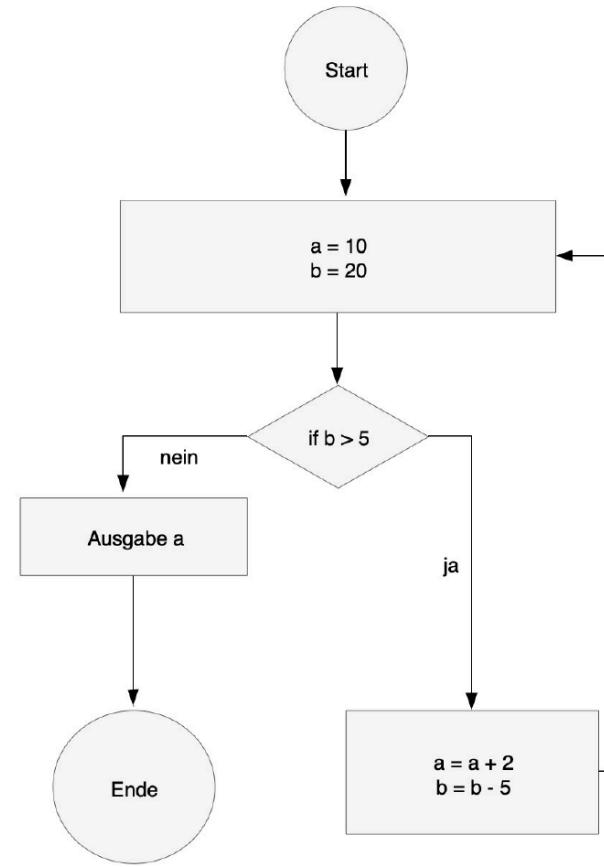
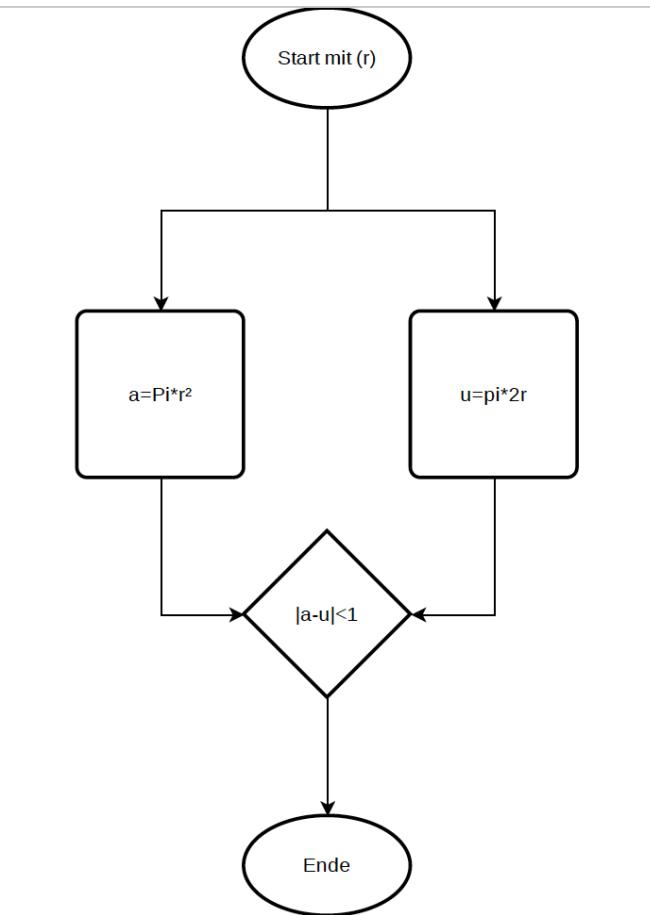
Bedingung

if-Anweisung



Flussdiagramme





Datentypen

Primitiven Typen:

- **Number**
- **String**
- **Boolean** : Wahrheitswert, kann wahr (**true**) oder falsch (**false**)
- **Null**

Beim Datentyp Null handelt es sich um ein Objekt mit dem Wert null. Dieser Datentyp weist auf einen leeren Wert hin. Wird einer bereits deklarierten und definierten Variablen der Wert null zugewiesen, bleibt die Variable für die weitere Verwendung erhalten. Unter anderem lassen sich so fließende Programmabläufe starten und stoppen.

- **Undefined**

Dieser Datentyp weist darauf hin, dass mit einer Variablen gearbeitet, die nur deklariert aber nicht definiert wurde oder etwas nicht Vorhandenes abgefragt wird, wie eine nicht vorhandene Eigenschaft eines Objektes.

Objekt-Typen:

- **Object**

Beispiel

```
var typ1 = "Ich bin ein String!";
var typ2 = 432.02;
var typ3 = false;
var typ4 = null;
var typ5, typ;
```

Boolean : Wahrheitswert

Boolesche Werte (Zustände)

- **Wahr (true oder 1)**
- **Falsch (false oder 0)**

boolesche Operatoren (Aussagenverknüpfungen)

- **und (&&) (Konjunktion)**
- **oder (| |) (Disjunktion)**
- **genau dann, wenn (==) (Äquivalenz)**
- **nicht (!) (Negation)**

if-Syntax

```
if (Bedingung)  
{ Anweisungen; }  
else  
{ Anweisungen; }
```

Beispiel:

```
if (zahl < 999 && zahl > 99)
{ alert(„die Zahl ist dreistellig“); }
else
{ alert(„Eine neue Zahl eingeben“); }
```

Wenn (zahl < 999 && zahl > 99)
true ist, dann

Wenn (zahl < 999 && zahl > 99)
false ist, dann

Beispiel:

```
if (x <= 0 || x > 10)
```

```
{y = 0;}
```

Wenn $x \leq 0$ ist oder $x > 10$, dann soll y gleich 0 gesetzt werden

```
else
```

```
{y = x * 10;}
```

Sonst wird y gleich x mal 10 gesetzt

Welcher Wert bekommt y für x=3? 30

Welcher Wert bekommt y für x=-3? 0

Vergleichsoperatoren

Operator	Bedeutung	Ergebnis der Operation
<code>==</code>	ist gleich	<code>true</code> , wenn die Werte gleich sind, sonst <code>false</code> .
<code>!=</code>	ungleich	<code>true</code> , wenn die Werte ungleich sind, sonst <code>false</code> .
<code>></code>	größer	<code>true</code> , wenn der linke Wert größer als der rechte ist, sonst <code>false</code> .
<code>>=</code>	größergleich	<code>true</code> , wenn der linke Wert größer oder gleich dem rechten ist, sonst <code>false</code> .
<code><</code>	kleiner	<code>true</code> , wenn der linke Wert kleiner als der rechte ist, sonst <code>false</code> .
<code><=</code>	kleinergleich	<code>true</code> , wenn der linke Wert kleiner oder gleich dem rechten ist, sonst <code>false</code> .

Wahrheitstabellen für logische Ausdrücke

Aussagen in der Aussagenlogik können nur *wahr* oder *falsch* sein

A	B	A und B
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	falsch
falsch	falsch	falsch

Belegungen für die Teilaussage A

Belegungen für die Teilaussage B

resultierende Wahrheitswerte für die zusammengesetzte Aussage "A und B"

resultierender Wahrheitswert für die Belegung A wahr und B wahr

resultierender Wahrheitswert für die Belegung A falsch und B wahr

Wahrheitstabellen für logische Ausdrücke

P	q	$P \& q$	$P \vee q$	$P = q$	$\neg P$
1	1	1	1	1	0
1	0	0	1	0	0
0	1	0	1	0	1
0	0	0	0	1	1

P	q	$P \& q$	$P \vee q$	$P = q$	$\neg P$
t	t	t	t	t	f
t	f	f	t	f	f
f	t	f	t	f	t
f	f	f	f	t	t

Priorität der Booleschen Ausdrücke

Allgemein gesprochen verwendet Javascript bei booleschen (logischen) Ausdrücken ähnliche Regeln wie bei arithmetischen Ausdrücken (also Rechnungen mit +, -, *, /)

Dabei gelten folgende Regeln:

Operator	Priorität	entspricht
!	3	unäres - (wie bei -3)
&&	2	*, /
	1	+, -

Wie wird dieses Ausdruck berechnet ? a || b && !c || d

(a || (b && (!c))) || d

Javascript Funktionen

```
if (isNaN(x))  
{ console.log ("Das ist keine Zahl"); }  
  
if (Math.ceil(x) >5)  
{document.getElementById("output").innerHTML = "Nicht bestanden!";}
```

<https://www.mediaevent.de/javascript/math-zahlen-numbers.html>