

## Abteilung IT-Aus- und Weiterbildung

### Konstanten-, Variablen- und Methodendeklarationen

- erstes Zeichen ein Buchstabe
- folgende Zeichen Buchstaben, Zahlen und einige Sonderzeichen
- keine Leerzeichen
- immer gleiche Regeln anwenden (konsistenter Benennungsstil)
- z.Bsp:

```
const additionalRabatt;  
function calculateCallsForAgent() {};
```

### Datentypen

	Typ	Typbezeichnung JavaScript
	Text	string
	Zahl	number
	Wahrheitswert	boolean
	Methoden	function
	undefiniert	undefined
	alle anderen (Window, Dokument, Browser, reguläre Ausdrücke, Arrays, der Wert null, etc.)	object

Notizen:

## Abteilung IT-Aus- und Weiterbildung

Sprachelemente			
Bezeichnung	Beschreibung	Syntax	Beispiel
Konstantendeklaration	ein konstanter Wert soll im Programm mit einem Namen definiert werden	<code>const [Variablenname] [ = Wert];</code>	<code>const MehrwertSteuer = 0.19;</code>
Variablendeklaration	ein variabler Wert soll im Programm mit einem Namen definiert werden	<code>var [Variablenname] [ = Wert]</code>	<code>var CallCenterAgent = 'Susi Calling';</code>
Zuweisung	einer Variablen soll ein Wert zugewiesen werden	<code>=</code>	<code>CallCenterAgent = 'Jens Caller';</code>
Vergleich	eine Variable soll auf ihren Inhalt geprüft werden	<code>==, !=, ===, !==, &gt;, &gt;=, &lt;, &lt;=</code>	<code>var KontoStand &gt; 1200;</code>
Einfache Fallunterscheidung	wenn eine Variable den Wert x hat, soll Anweisung(en)_1 ausgeführt werden, ansonsten Anweisung(en)_2	<pre>if([Bedingung]) {     AnweisungsBlock_1; } [else {     AnweisungsBlock_2; }];</pre>	<pre>if(KontoStand &lt; 1200) {     var KontofuehrungsGebueren = 10; } else {     var KontofuehrungsGebueren = 0; }</pre>
Mehrfache Fallunterscheidung	Entsprechend des Wertes einer Variablen soll bei Wert1 Anweisung(en)_1, bei Wert2 Anweisung(en)_2 ausgeführt werden, usw.	<pre>switch([Variable]){     case [Wert1]: [Anweisungsblock1];                 break;     case [Wert2]: [Anweisungsblock2];                 break;     default:         [Anweisungsblock_ansonsten];                 break; };</pre>	<pre>switch(TagDerWoche){     case 1: var WeekDay = 'Montag';             break;     case 2: var WeekDay = 'Dienstag';             break;     default: var WeekDay = 'n. klar';             break; };</pre>

## Abteilung IT-Aus- und Weiterbildung

Zählergesteuerte Schleife (Numerische Schleife)	Eine Anweisung soll n-mal ausgeführt werden	<pre>for([Zaehler=Startwert];     [Abbruchbedingung];     [Zaehlerinkrement]) {     AnweisungsBlock; }</pre>	<pre>for(var i=0; i &lt; AnzMitarbeiter; i++) {     checkeCalls(Mitarbeiter); }</pre>
Kopfgesteuerte Schleife	solange eine Bedingung zutrifft soll ein Anweisungsblock mehrfach ausgeführt werden; ACHTUNG: innerhalb der Schleife muss eine Abbruchbedingung gesetzt werden	<pre>while([Bedingung]) {     AnweisungsBlock; }</pre>	<pre>while(AktuellerMitarbeiterZaehler &lt; AnzMitarbeiter) {      calculateCalls(MitarbeiterNummer);     AktuellerMitarbeiterZaehler++; }</pre>
Fußgesteuerte Schleife	Führe eine Anweisung durch und wiederhole diese solange bis eine Bedingung zutrifft; ACHTUNG: innerhalb der Schleife muss eine Abbruchbedingung gesetzt werden	<pre>do {     Anweisungsblock; } while ([Bedingung]);</pre>	<pre>do {     Anweisungsblock; } while (CurrentCounter &lt; AnzMitarbeiter);</pre>
Eigene Methode bilden	für die Übersichtlichkeit des Programmcodes soll ein Block von Anweisungen als eigene Methode zusammengefasst werden	<pre>function FunctionName(Parameter) {     AnweisungsBlock;     return Resultat };</pre>	<pre>function HoleCalls(Mitarbeiter) {     var getCalls = lookForCalls();     return getCalls; }</pre>
Meldung ausgeben	Es soll im Programm eine Meldung ausgegeben werden	<pre>alert(MeldungsText);</pre>	<pre>alert('Das Programm ist beendet!');</pre>
Wert eingeben	Es soll im Programmverlauf ein Wert eingegeben werden	<pre>InputVariable = prompt([Text]);</pre>	<pre>CountRepeats = prompt('Bitte Anzahl der Wiederholungen angeben!');</pre>
Notizen:			

## Abteilung IT-Aus- und Weiterbildung

Beim Auftreten eines Fehlers soll das Programm sinnvoll reagieren	Eine Fehlerbehandlungsroutine soll gebildet werden	<pre>try {   [Bedingung1] throw [Text1];   [Bedingung1] throw [Text1];   ...   catch (e) {     e === Text1 doSomething;     e === Text2 doSomethingElse;     ...   } }</pre>	<pre>try {   if (value === 0) {     throw "division by zero";   } else {     throw "okay";   } } catch (e) {   if (e === "okay") {     print(e);     return;   } else {     print(e);     return;   } }</pre>
Programmablauf unterbrechen	eine Programm-struktur soll vorzeitig verlassen werden	<pre>{   Anweisung1   break;   Anweisung2 }</pre>	<pre>/* Endlosschleife */ for (var counter=0;;counter++) {   document.write(counter + '&lt;br&gt;');   if (counter == 100) break; }</pre>
Den Programmcode dokumentieren	Wichtige Teile der Programmlogik sollen dokumentiert werden	<pre>/* Kommentar */ // Kommentar</pre>	<pre>/* alle Callcenteragents sollen nacheinander aufgerufen werden */</pre>