

PB notes 03 theory

1 strings bouwen met string interpolatie

string interpolatie => \$ voor de **string literal** zetten => in de string tussen {...} een <var> plaatsen vb.:

```
1 string naam = "jan";
2 int leeftijd = 34;
3
4 string bericht = $"{naam} is {leeftijd} jaar oud.";
5 Console.Write(bericht);
```

als je in deze string literal met interpolatie een } wil hebben, moet je deze verdubbelen }}

2 werken met getallen

2.1 rondom getallen

om random getallen te bekomen moet je eerst deze initialiseren met deze lijn:

```
1 Random <rnd> = new Random();
```



om random getal te krijgen = <rnd>.Next() :

```
1 Random <rnd> = new Random();
2 int <randomInt> = <rnd>.Next();
```

bv.:

```
1 Random rnd = new Random();
2 int worp1 = rnd.Next(1, 7);
3 int worp2 = rnd.Next(1, 7);
4
5 Console.WriteLine(worp1);
6 Console.WriteLine(worp2);
```

2.2 Math functionaliteit

Math klasse => bevat veel **functies** die wiskundige berekeningen makkelijker maken. official documentation:

- <https://docs.microsoft.com/en-us/dotnet/api/system.math>

functie	uitleg
<code>Math.Min(<var1>, <var2>)</code>	neemt de kleinste waarde tussen <code><var1></code> en <code><var2></code>
<code>Math.Max(<var1>, <var2>)</code>	neemt de grootste waarde tussen <code><var1></code> en <code><var2></code>
<code>Math.PI</code>	de waarde van Pi = 3.14159...
<code>Math.Pow(<double1>, <double2>)</code>	resultaat is <code><double1></code> tot de macht van <code><double2></code>
<code>Math.Sqrt(<double>)</code>	berekent de vierkantswortel van <code><double></code>
<code>Math.Abs(<double>)</code>	geeft de absolute waarde van <code><double></code>

2.3 verkorte notatie voor eenvoudige berekening

voluit	verkort
<code><var> = <var> + 1</code>	<code><var>++</code>
<code><var> = <var> - 1</code>	<code><var>--</code>
<code><var1> = <var1> + <var2></code>	<code><var1> += <var2></code>
<code><var1> = <var1> - <var2></code>	<code><var1> -= <var2></code>
<code><var1> = <var1> * <var2></code>	<code><var1> *= <var2></code>
<code><var1> = <var1> / <var2></code>	<code><var1> /= <var2></code>

3 Variabelen en scope

Declaratie van een variabele:

- introduceert variabele
- geeft deze een naam
- geeft ze een datatype => compiler maakt code tijdens uitvoering die een opslagplaats voorziet in het geheugen vb.:

```
1  int i = 12;
2  double d = 34.56;
3  string s = "Hello world!";
```

3.1 Naamgeving en declaraties

Regels rond **naamgeving** van **variabelen**

- variabelen **starten** met een **Lower Case**
- **camelCasing**: als naam uit meerdere woorden bestaat => alles behalve 1ste woord begint met een Upper Case
- **nooit** een **prefix** die het **datatype** aanduidt van de variabele
- introduceer enkel **1 variabele per lijn**
- **CONSEQUENTIE!!!!**

3.2 Scope

scope of a `<var>` defines where the `<var>` can be accessed/modified. **Scope** holds the current state of variables along with their respective values. Scope of a `<var>` starts with the declaration and ends with the end of the codeblock/scope

categories of scope

- **Block** level scope
 - generally in codeblock of if/else, for, while statements
 - limited to the codeblocks they are declared in
 - These variables are also termed as the loop variables or statements variable as they have limited their scope up to the body of the statement in which it declared.
- **method** level scope method = the block of codes or statements in a program that gives the user the ability to **reuse** the same code
- **class** level scope blablabla

much better explanation: <https://www.geeksforgeeks.org/scope-of-a-variable/>

3.3 Const

`const <var>` = variable that **cannot get a new value** after declaration (Almost) ALWAYS use `const <var>` instead of **magic values** to give clarity to your code. Not if making it a `const <var>` if:

- does not make it more clear
- value only is used once
- can't think of a meaningful name
- the value will ever change
- `const` is short for **constant**

magic values = hardcoded values that are used in code bad practice! not clear where the values come from or what they mean if ever needed to change = lot of work to change every value independently => prone to get bugs

4 Traceertabellen

= tabel wat is deze info?? als je Visual studio runt met break or steps enabled kan je ook een "traceertabel" zien. => Locals / Autos tab

"Pasted image 20241004212436.png" could not be found.