

SOFTWARE ONTWIKKELEN

AGENDA VOOR VANDAAG

- Lesdoelen
- Lastige opdrachten?
- Herhalingsstructuren - For lussen
- Herhalingsstructuren - Foreach lussen
- Debugger
- Leerstrategie: verwerking

LESDOELEN

Na de les van vandaag:

- Begrijp je het concept 'Loops' en kun je deze toepassen in C#
- Kun je de debugger gebruiken in Visual Studio Code om te 'debuggen'.
- Kun je de leerstrategie 'verwerking' inzetten voor het beter begrijpen van loops



WELKE OPDRACHTEN WAREN LASTIG?

FOR LUSSEN

Een for-lus is een herhalingsstructuur die **dezelfde** code uitvoert zolang aan een bepaalde voorwaarde wordt voldaan.

Waarom kan dit handig zijn? Kun je een voorbeeld bedenken?



FOR LUSSEN

```
for (initialisatie; conditie; iteratie)
{
    // Code die herhaald wordt
}
```

```
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine(i);
}
```

FOR LUSSEN - OEFENING

Als je in 1 jaar elke maand 100,- spaart, hoeveel heb je dan na een jaar?

Hoe zou je dit programmeren?

FOR LUSSEN – OEFENING

Schrijf een for-lus die de even getallen tussen 2 en 10 print.

Hoe zou je dit programmeren?

FOREACH LUSSEN

Een **foreach-lus** wordt gebruikt om eenvoudig door collecties (zoals arrays of lijsten) te itereren.

Makkelijker te gebruiken dan for-lussen voor het doorlopen van elke waarde in een collectie zonder de index handmatig te beheren.

FOREACH LUSSEN

```
foreach (var item in collectie)
{
    // Code die wordt uitgevoerd voor elk item in de collectie
}
```


FOREACH LUSSEN

```
string[] namen = ["Alice", "Bob", "Charlie"];  
  
foreach (string naam in namen)  
{  
    Console.WriteLine("Hallo " + naam);  
}
```

Hallo Alice

Hallo Bob

Hallo Charlie

FOREACH OEFENING

Maak een programma dat alle even getallen uit een lijst van getallen print.

```
// Zo declareer je een variabele met als datatype een array (lijst) met getallen  
int[] lijstMetGetallen = [10,15,120,25,30];
```

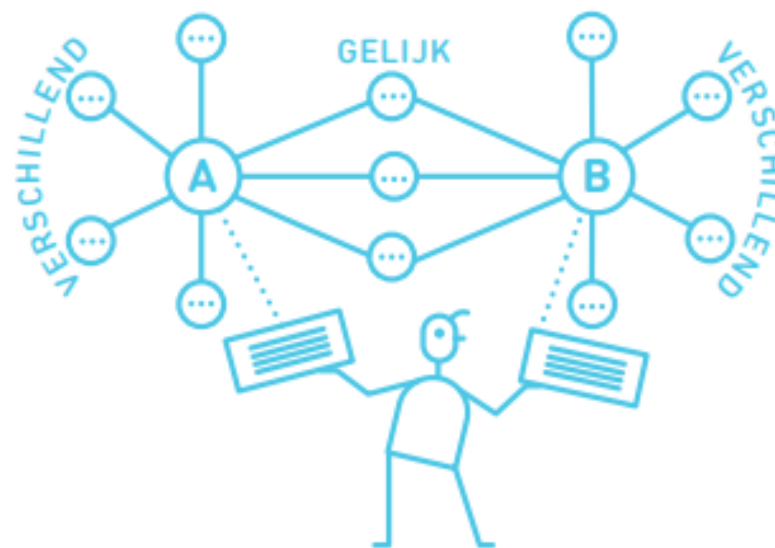

DEBUGGER

Een onderdeel van je IDE die je helpt bij het analyseren van je code tijdens het uitvoeren.

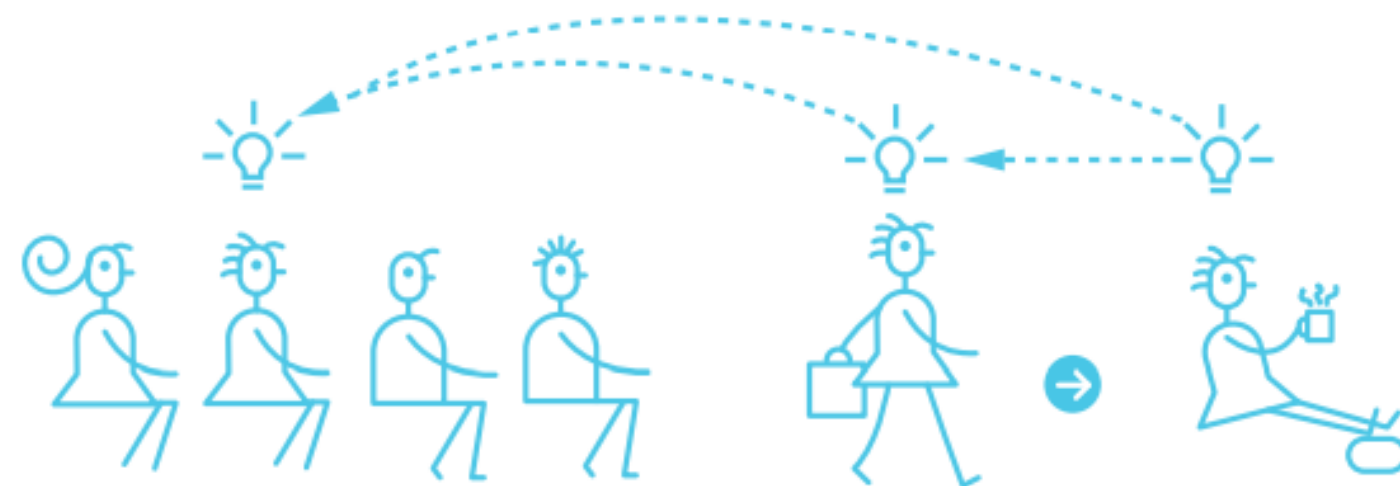
- Breakpoints
- Watch

LEERSTRATEGIEËN

LEARNINGSIENTISTS.ORG



Wat is *verwerking* ook alweer?



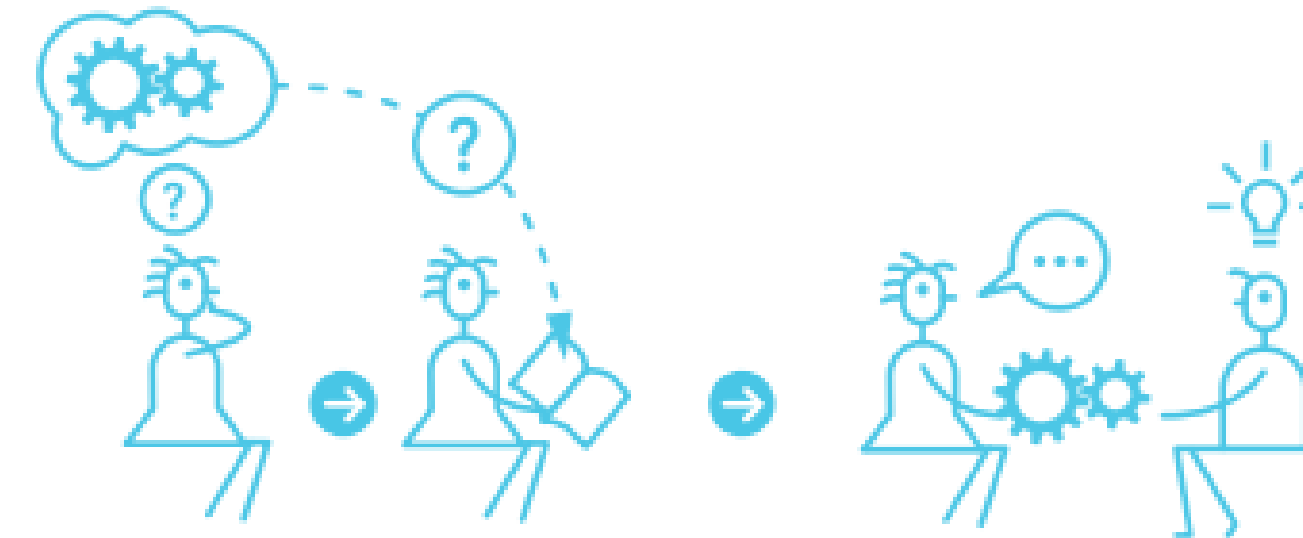
WAT IS VERWERKING OOK ALWEER?

LEARNINGSUBJECTS.ORG

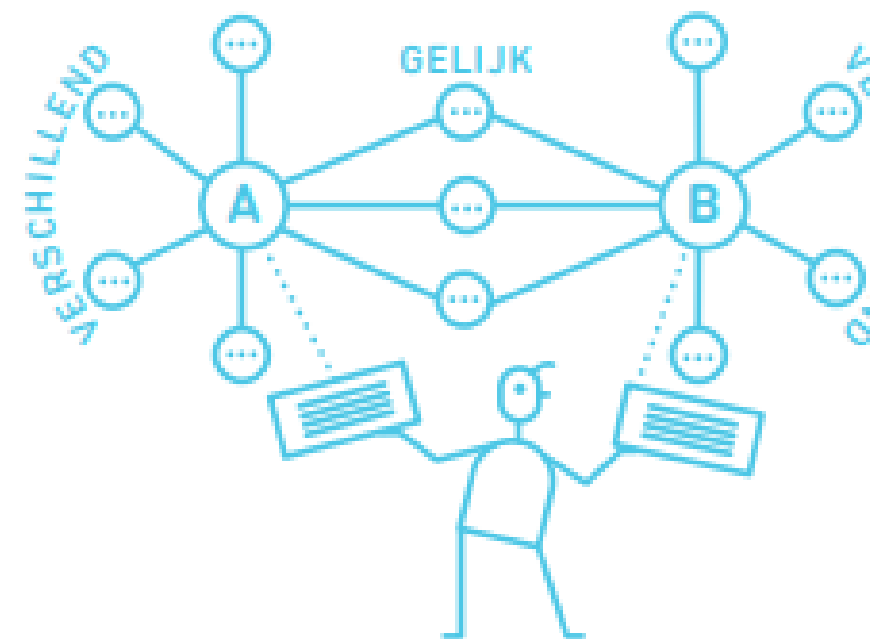


HOE DOE JE HET?

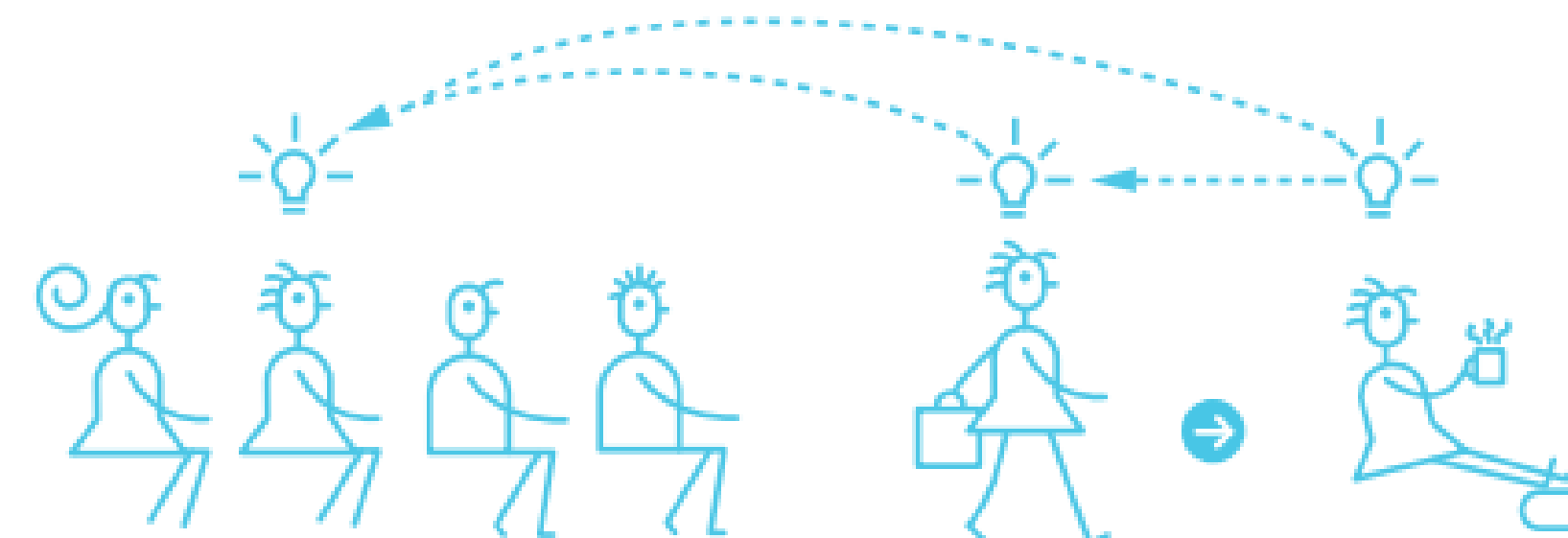
Stel jezelf vragen tijdens het studeren over hoe dingen in elkaar zitten en waarom ze zo werken. Zoek vervolgens de antwoorden op in je studiemateriaal en bespreek de antwoorden met je klasgenoten.



Terwijl je het studiemateriaal verwerkt, leg je verbanden tussen verschillende ideeën om uit te leggen hoe ze verbonden zijn en op elkaar inwerken. Neem twee ideeën en onderzoek hoe ze verschillen en hoe ze op elkaar lijken.



Beschrijf hoe de leerstof overeenkomt of verschilt met eigen ervaringen en herinneringen. Probeer in de loop van de dag zelf verbanden te leggen met wat je leert op school.



VERWERKING – DE DOCENT DOET HET VOOR

1. [de vraag]
2. [het linkje]
3. [de verbanden, misschien met plaatje?]

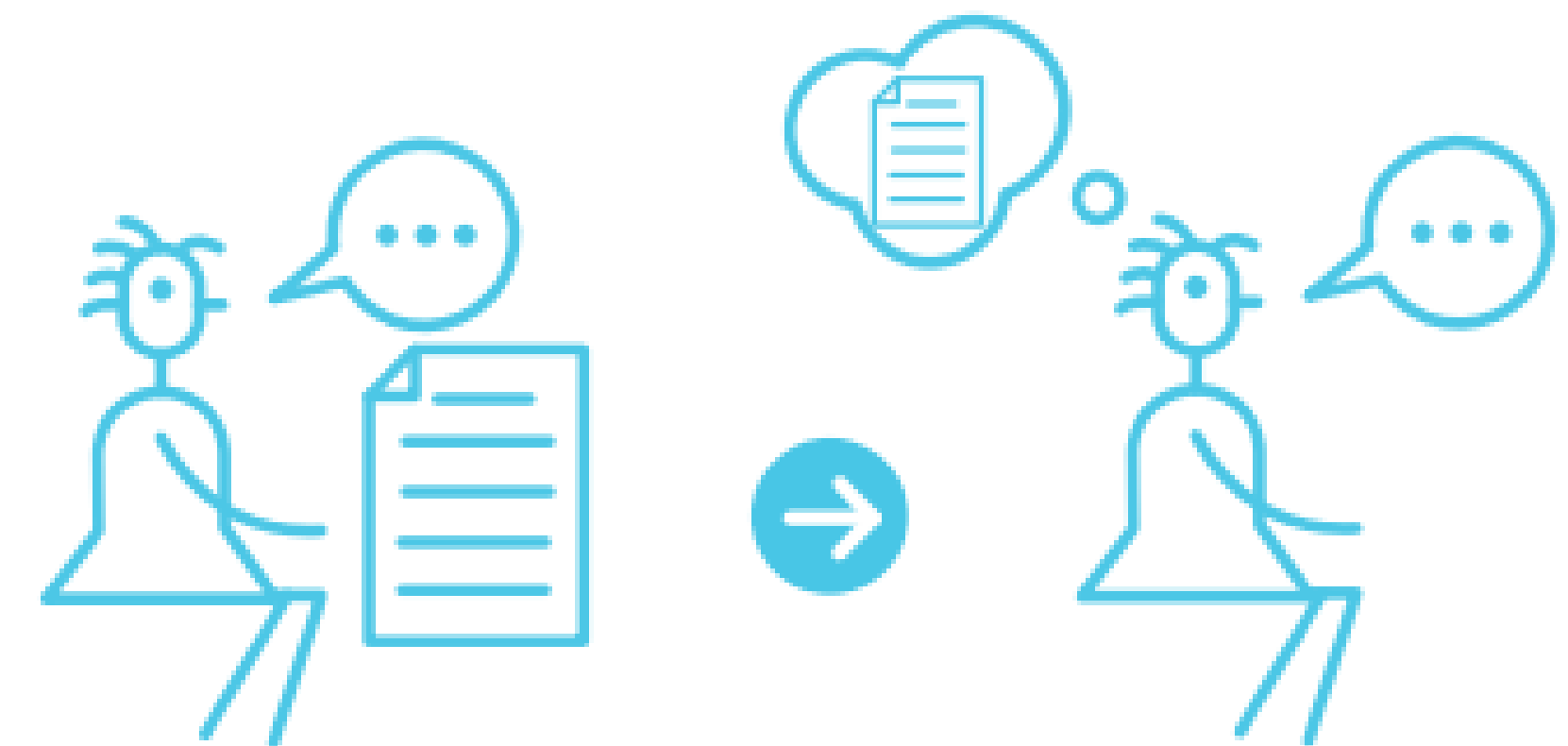
VERWERKING OEFENEN (20 MINUTEN)

1. Schrijf een vraag op over loops
2. Zoek het antwoord op je vraag in de lesstof op en bestudeer dit (3 min)
3. Leg **in eigen woorden** aan een klasgenoot uit wat je vraag was en wat het antwoord hierop is, met voorbeelden uit oefeningen die je gemaakt hebt (5 min)
4. Controleer samen of je het nu goed hebt uitgelegd (2 min)
5. Draai de rollen om



VERWERKING OEFENEN

- Probeer vanavond thuis nog eens hetzelfde antwoord op de vraag die je vandaag gesteld hebt te geven. Leg het uit aan een huisgenoot (of eendje).
 - Weet je alles nog?
 - Probeer een steeds beter antwoord te formuleren



LESDOELEN BEHAALD?

Bewaar de vraag die je jezelf gesteld had goed! Neem deze mee naar de volgende les!

ASYNCHROON

- Herhalingsstructuren - While lussen
- C# functionaliteiten (Math, Random)