

## Opgave 1: "Try It Yourself"

### Section 1 - Understanding Prompts

#### Step 2:

#### 1. Explain the proces of photosynthesis in simple terms:

Sure! 🌱 Here's a simple explanation of **photosynthesis**:

Photosynthesis is the way green plants make their own food.

They use **sunlight**, **water**, and **carbon dioxide** from the air.

Here's how it works step by step:

1. ☀️ The **sun** shines on the plant's leaves.
2. 🌱 Inside the leaves are tiny parts called **chloroplasts**, which contain **chlorophyll** — a green substance that captures sunlight.
3. 💧 The plant's **roots** take in **water** from the soil.
4. 🌬️ The **leaves** take in **carbon dioxide** from the air.
5. Using the energy from sunlight, the plant **mixes** water and carbon dioxide to make **sugar** (its food).
6. 🌬️ The plant also releases **oxygen** into the air — which humans and animals need to breathe.

So in short:

☀️ **Sunlight** + **Water** + **Carbon Dioxide** → **Sugar** + **Oxygen**

That's photosynthesis — how plants eat light and make life possible for all of us! 🌱❤️

#### 2. List 10 keywords and their definitions relevant to photosynthesis:

List 10 keywords and their definitions relevant to photosynthesis.

Here are **10 keywords** and their **definitions** related to **photosynthesis** 🌱 :

1. **Photosynthesis** – The process by which green plants use sunlight to make food from water and carbon dioxide.
2. **Chlorophyll** – The green pigment in plants that absorbs sunlight and helps turn it into energy.
3. **Chloroplasts** – Tiny parts inside plant cells where photosynthesis happens.
4. **Carbon Dioxide (CO<sub>2</sub>)** – A gas in the air that plants take in through their leaves to use in photosynthesis.
5. **Water (H<sub>2</sub>O)** – Taken up by the plant's roots from the soil and used along with carbon dioxide to make food.
6. **Sunlight** – The energy from the sun that powers photosynthesis.
7. **Glucose** – The sugar that plants make during photosynthesis; it's their main source of energy and food.
8. **Oxygen (O<sub>2</sub>)** – A gas that plants release into the air during photosynthesis; humans and animals need it to breathe.
9. **Stomata** – Tiny openings on the underside of leaves that let gases like carbon dioxide and oxygen move in and out.
10. **Energy** – What plants get from sunlight and use to change carbon dioxide and water into glucose and oxygen.

Would you like me to write the **outline for the short** ⬇️📺 (Step 3) next? 😊

### 3. Write an outline for a short video about photosynthesis.

Write an outline for a short video about photosynthesis.

Here's a clear and engaging **outline for a short video about photosynthesis** 🌱 📺

**Title:**

"How Plants Eat Sunshine!"

**Length:**

About 30 seconds – 1 minute

**Outline:**

1. Opening Scene (0–5 seconds)

- Bright sunshine over a garden full of green plants.
- Narrator says: "Have you ever wondered how plants eat?"

## 2. Meet the Leaf (5–10 seconds)

- Zoom in on a smiling cartoon leaf character.
  - Leaf says: "Hi! I'm Lily the Leaf, and I make food using sunlight!"
- 

## 3. Gathering Ingredients (10–20 seconds)

- Fun animations show:
    - Roots sipping **water** from the soil 💧
    - Leaf breathing in **carbon dioxide** from the air 🌬️
    - Sunbeams shining down ☀️
  - Narrator: "Plants mix sunlight, water, and carbon dioxide to make food!"
- 

## 4. The Magic Inside (20–25 seconds)

- Inside the leaf, **chloroplasts** glow as they turn sunlight into **energy**.
  - Simple text appears: "Sunlight + Water + Carbon Dioxide = Food + Oxygen"
- 

## 5. The Result (25–30 seconds)

- The plant looks happy and strong.
  - Oxygen bubbles float into the air.
  - Narrator: "That's photosynthesis — how plants make food and give us air to breathe!"
- 

## 6. Ending (30–35 seconds)


- Leaf waves goodbye: "Thanks for learning with me! Keep growing bright!" 🌱 ✨
- 

Would you like me to now write the **screenplay** version (as in Step 3: "Act as a screenwriter...") for 8-year-olds?

## Step 3. Apply the elements of a good prompt to craft your own:

"Act as a screenwriter. I want you to write a screenplay for 8-year-old students to learn about photosynthesis. Your answer should be written in a happy and child-friendly tone, and long enough for a 30-second video."

Here's a **child-friendly 30-second screenplay** about **photosynthesis**, written in a bright, fun tone 🌱 🎬

 **Title: "How Plants Eat Sunshine!"**

**[Scene 1 – Sunny Garden, cheerful music playing]**

**Narrator:** (excited)

"Hey kids! Did you know plants can make their own food?"

**[Cut to a smiling cartoon leaf waving]**

**Leaf:** "Hi there! I'm Lily the Leaf, and I'm a super chef!"

**[Scene 2 – Close-up of Lily with sun rays shining]**

**Leaf:** "Every day, I use three yummy ingredients: sunlight, water, and carbon dioxide!"

**[Animation shows sunlight beams, drops of water from the roots, and bubbles of air coming in]**

**Narrator:** "The sunlight gives Lily energy. The water comes from her roots. And carbon dioxide comes from the air!"

**[Scene 3 – Inside the leaf, glowing green chloroplasts sparkle]**

**Leaf:** "Inside me, tiny helpers called chloroplasts mix it all together to make delicious plant food called glucose!"

**[Scene 4 – The plant stands tall and strong; oxygen bubbles float away]**

**Narrator:** "And guess what? Plants share something special back with us — oxygen to breathe!"

**Leaf:** (giggles) "So every breath you take, remember to thank a plant!"

[Scene 5 – Ending, upbeat music]

**Narrator:** “That’s photosynthesis — the magic recipe of life!”

**Leaf:** “Keep growing bright, just like me!” 🌻 🌿 ✨

---

Would you like me to expand this into the **3-minute version for 15-year-olds** (Step 4)?

Step 4: Reiterate the prompt so that the output is for a 3 minute educational video for 15-year-old students:

Reiterate the prompt so that the output is for a 3-minute educational video for 15-year-old students.

Absolutely! 🌱 Here’s the **revised version** of the same task — written as a **3-minute educational screenplay** for **15-year-old students**, using a slightly more mature, curious, and informative tone.

---

🎬 **Title: “The Power of Photosynthesis – How Plants Turn Light into Life”**

[Scene 1 – Opening, 0:00–0:30]

*(Calm music, sunlight over a forest. A soft breeze moves through the trees.)*

**Narrator:**

“Every breath you take... begins with light.

But how exactly do plants turn sunlight into the oxygen we breathe and the food they need to live?

Let’s dive into the amazing process of photosynthesis.”

**[Scene 2 – The Big Picture, 0:30–1:00]**

*(Animation zooms in on a single leaf.)*

**Narrator:**

"Photosynthesis happens inside special parts of plant cells called **chloroplasts**.

These contain **chlorophyll**, the green pigment that captures energy from sunlight."

*(Chlorophyll molecules sparkle as light hits them.)*

**Narrator:**

"When sunlight hits chlorophyll, it gives the plant energy — energy it uses to power a chemical reaction."

---

**[Scene 3 – The Ingredients, 1:00–1:30]**

*(Graphics show roots taking in water and leaves absorbing carbon dioxide.)*

**Narrator:**

"Plants pull **water** from the soil through their roots and **carbon dioxide** from the air through tiny pores called **stomata**.

With sunlight as energy, they combine these ingredients to make something new."

---

**[Scene 4 – The Chemical Magic, 1:30–2:00]**

*(Inside the leaf: atoms and molecules move, forming glucose and oxygen.)*

**Narrator:**

"Here's the simple formula:

**Carbon dioxide + Water + Light energy → Glucose + Oxygen**

Glucose is a type of sugar that stores energy for the plant.

Oxygen is released into the air — and that's what keeps animals and humans alive."

---

**[Scene 5 – Why It Matters, 2:00–2:30]**

*(Footage of forests, crops, and underwater plants.)*

**Narrator:**

"Photosynthesis doesn't just feed plants — it fuels our entire planet.

It's the foundation of every food chain and the reason Earth has breathable air."

[Scene 6 – Real-World Connection, 2:30–2:50]

*(Students doing experiments, growing plants under lights.)*

**Narrator:**

“Scientists study photosynthesis to design cleaner energy sources — like artificial leaves that could one day produce fuel from sunlight and water.”

---

[Scene 7 – Conclusion, 2:50–3:00]

*(Return to the sunny forest. Text appears: “Photosynthesis = Light into Life.”)*

**Narrator:**

“So next time you see a leaf glowing in the sun, remember — it’s not just green.  
It’s working hard to keep the planet alive.” 🌍💚

---

Would you like me to format this version as a **script ready for voice-over recording** (with estimated word timing and narrator cues)?

## Step 4 continued:

Reiterate the prompt so that the output is for a 2-minute educational video for 6-year-old students.

Perfect 😊 — here’s the **reiterated version** of the prompt, adjusted for a **2-minute educational video** aimed at **6-year-old students**, written in a **happy, simple, and story-like tone** that’s easy to follow.

---

🎬 **Title: “The Magic of Sunshine Food!”**

[Scene 1 – Opening, 0:00–0:20]

*(Bright music. A sunny garden with colorful flowers and a friendly green leaf waving.)*

**Narrator:**

“Hi there, little explorers! Do you know how plants eat?

They don’t have mouths like we do — they make their own food from sunlight!

Let’s see how they do it!”

**[Scene 2 – Meet Sunny the Leaf, 0:20–0:40]**

*(A smiling cartoon leaf appears.)*

**Sunny the Leaf:**

"Hello friends! I'm Sunny! Every day, I make my own food using a little magic from the sun!"

*(Sunlight sparkles and shines on Sunny's face.)*

**Sunny:**

"I need three things: sunlight, water, and air. Let's collect them together!"

---

**[Scene 3 – Gathering Ingredients, 0:40–1:10]**

*(Fun animation sequence.)*

- **Water** travels up from the roots — little blue drops bounce along.
- **Air** bubbles with **carbon dioxide** float into the leaf.
- **Sunlight** beams down with happy faces.

**Narrator:**

"When Sunny mixes sunlight, water, and air, something amazing happens inside her leaves!"

---

**[Scene 4 – Inside the Leaf, 1:10–1:30]**

*(Zoom into glowing green cells full of dancing chloroplasts.)*

**Sunny:**

"My tiny helpers, the chloroplasts, work hard to turn all that into food called glucose!"

**Narrator:**

"That's the energy Sunny uses to grow big and strong."

---

**[Scene 5 – A Gift for Us, 1:30–1:50]**

*(Oxygen bubbles float out of the leaf toward happy children and animals.)*

**Narrator:**

"And guess what? While Sunny makes her food, she also makes oxygen for us to breathe!"

**Sunny:**

"So every breath you take, say 'thank you, plants!'" 🌿💧

---



---

[Scene 6 – Ending, 1:50–2:00]

*(Upbeat music, all the plants wave.)*

**Narrator:**

"That's called **photosynthesis** — the way plants eat sunshine and share life with the world!"

**Sunny:**

"Stay bright, little scientists!" 🌻 🌱

---

Would you like me to add **narration timing and voice-over pacing** (e.g., seconds per line) so it fits exactly into a 2-minute recording?

## Step 4 continued:

Reiterate the prompt so that the output is for a 3-minute educational video for university students.

Excellent 🌱 — here's the **reiterated prompt** and the resulting **3-minute educational video script** designed for **university-level students**.

This version keeps a professional, clear, and intellectually engaging tone while staying visually dynamic and educational.

---

🎓 **Title: "Photosynthesis – Converting Light into Chemical Energy"**

---

**[Scene 1 – Introduction, 0:00–0:30]**

*(Opening shot: sunlight over a forest canopy, transitioning to microscopic visuals of leaf cells.)*

**Narrator (calm, academic tone):**

"Photosynthesis is one of the most essential biochemical processes on Earth.

It transforms light energy into chemical energy — sustaining nearly all forms of life.

In this video, we'll explore how plants, algae, and some bacteria perform this remarkable energy conversion."

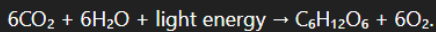
---

**[Scene 2 – The Overview, 0:30–1:00]**

*(Animated diagram of a leaf showing inputs and outputs.)*

**Narrator:**

"The general equation is:



In words: carbon dioxide and water, using sunlight, are converted into glucose and oxygen.

This process occurs primarily in the **chloroplasts**, specialized organelles found in plant cells."

**[Scene 3 – The Light-Dependent Reactions, 1:00–1:30]**

*(Inside the chloroplast: the thylakoid membrane glows with electron flow animations.)*

**Narrator:**

"The first stage, the **light-dependent reactions**, takes place in the **thylakoid membranes**.

When photons strike chlorophyll, electrons become excited and move through the **electron transport chain**.

This drives the formation of **ATP** and **NADPH** — two energy carriers essential for the next stage."

---

**[Scene 4 – The Calvin Cycle, 1:30–2:00]**

*(Transition to the stroma of the chloroplast, showing a molecular loop animation.)*

**Narrator:**

"The second stage, the **Calvin Cycle**, occurs in the **stroma**.

Here, the energy stored in ATP and NADPH is used to fix **carbon dioxide** into organic molecules.

Through a series of enzyme-driven steps, primarily catalyzed by **RuBisCO**, carbon is transformed into **glucose**, which plants use for growth and storage."

[Scene 5 – Importance and Applications, 2:00–2:40]

*(Footage of forests, crops, and scientific labs working on bioengineering.)*

**Narrator:**

"Photosynthesis is the foundation of Earth's carbon and oxygen cycles.

It provides the energy source for heterotrophic organisms and regulates atmospheric gases.

Understanding its mechanisms has inspired fields such as **artificial photosynthesis**, **biofuels**, and **carbon capture technologies**, where scientists aim to replicate or enhance nature's efficiency."

---

[Scene 6 – Closing Thoughts, 2:40–3:00]

*(Zoom out from a leaf cell to a view of Earth from space.)*

**Narrator:**

"In essence, photosynthesis connects the microscopic world of molecules to the macroscopic web of life.

From a single photon to global ecosystems — it is the process that turns sunlight into survival."

*(Fade to text: "Photosynthesis: Light → Life → Energy.")*

---

Would you like me to convert this version into a **narration script with timestamps and visual cues** (for a real 3-minute video production storyboard)?

## Section 2 - Gina's Investigation

Compose an Outline

Step 1: Go to Copilot chat and try this prompt following the instructions below:

Write an outline for a newspaper article that highlights the consequences of plastic pollution on the sea turtle population. Please write in an informational tone and format as ideas. Please make this longer in length.

Sure! 🌱 Here's a **detailed, informational outline** for a **newspaper article** about the **consequences of plastic pollution on sea turtles** — written as a structured list of **ideas and sections**, in the tone and format of a professional news feature.

## Outline: "Choking Seas: How Plastic Pollution Threatens Sea Turtles"

### I. **Headline / Lead**

- *Choking Seas: Plastic Pollution's Deadly Toll on Sea Turtles*
- Lead hook: a vivid image of a turtle entangled in plastic or a beach strewn with debris, drawing readers in.

### II. **Introduction — Why It Matters**

- Presentation of the global plastic crisis in the oceans; mention that marine debris is one of the top threats to marine species.
- Statement: sea turtles are particularly vulnerable because they traverse both land (nesting) and sea (migration, feeding).
- Transition: "Let's examine how plastic affects these ancient mariners at every stage of life."

### III. Paths of Harm — How Plastic Affects Turtles

#### A. Ingestion (Eating Plastic Mistakenly)

- Turtles often mistake plastic bags and fragments for jellyfish or other prey.
- Ingestion leads to **intestinal blockages, internal injuries, perforations, and starvation.** [WWF Australia](#) +3
- Study: a quantitative analysis showed a **50% probability of mortality** once a turtle's gut contained 14 or more pieces of plastic. [PMC](#) +1
- Another study: ~86% of green turtles examined had plastic ingestion; ~14% showed entanglement. [ScienceDirect](#)
- Data: about 54% of examined sea turtles had ingested plastic debris in one study, with types of plastic varying by species. [SpringerLink](#)
- Sometimes plastic in the gut can mimic fullness, so turtles stop eating and waste away. [WWF Australia](#) +3

#### B. Entanglement

- Turtles become entangled in discarded fishing gear ("ghost nets"), ropes, plastic rings, and lines. [seaturtlestatus.o...](#) +4
- Consequences: restricted movement, inability to surface to breathe, lacerations, infections, drowning, starvation. [NOAA Fisheries](#) +3
- Example: more than 1,000 sea turtles estimated to die each year from entanglement alone. [seaturtlestatus.o...](#) +2
- Region-specific data: in Hawaii, fishing lines and hooks account for many injuries; many injured turtles are released alive after intervention. [NIST](#)

#### C. Habitat Degradation & Chemical Pollution

- Plastic debris can clog nesting beaches, alter sand structure or temperature, and interfere with hatchling emergence. [OUP Academic](#) +4
- Microplastics and plastic additives (e.g., BPA, heavy metals) leach toxic chemicals that can disrupt hormones, reproduction, immunity, or cause tumors. [Plastic Pollution ...](#) +3
- Some plastics absorb pollutants from seawater (PCBs, DDT, etc.), making them more toxic. [Wikipedia](#) +1
- Chemical exposure may feminize turtles (skewing sex ratios) or reduce fertility. [Plastic Pollution ...](#)

#### IV. Scale & Stage — When & Where Turtles Are Most Affected

- Hatchlings crawling across beaches may get caught in plastic on shore. [SEE Turtles +2](#)
  - Juvenile turtles in oceanic currents often overlap with “garbage patches” and floating debris zones. [Frontiers +3](#)
  - All seven species of sea turtles are impacted by ingestion and entanglement. [WWF Australia +2](#)
  - Regional studies: in one Red Sea study, 40% of examined turtles had ingested plastics. [PMC](#)
- 

#### V. Ecological & Population-Level Impacts

- Reduced survival rates lead to declining populations over time.
- Disruption of ecological roles: turtles help control jellyfish, maintain seagrass beds, and support coral reef health.
- Plastic effects can ripple through food webs, affecting fish and humans.
- Higher mortality in juvenile stages leads to fewer turtles reaching maturity and reproducing.

#### VI. Human & Societal Responsibility

- Plastic waste originates from land-based sources: litter, poor waste management, rivers, storm drains. [Sea Turtle Cons... +2](#)
  - Fishing industry’s lost, abandoned nets and gear contribute heavily. [seaturtlestatus.o... +3](#)
  - Coastal development, tourism, and beach litter amplify risks near nesting sites.
  - Global trade in plastic waste shifts pollution burdens.
- 

#### VII. Efforts, Solutions & Hope

- Conservation programs remove plastics, rescue & rehabilitate turtles, run beach cleanups.
- Policies banning single-use plastics, enforcing fishing regulations, and improving waste systems.
- Innovations: biodegradable materials, ghost-net recovery, filtration systems, artificial cleanup.
- Engaging local communities and education—behavior change at individual level matters.
- Success stories from regions with regulation and community efforts.

### VIII. Call to Action & Final Thought

- Encourage readers to reduce plastic use, support bans, donate to turtle conservation, volunteer, or advocate policy changes.
- Conclude with a reminder: "Plastic doesn't just pollute landscapes — it chokes lives in the ocean."

---

If you like, I can now **expand this outline into a full newspaper-style article** (~700 words) and include direct quotes or case studies with citations. Would you prefer that?

Experiment and Iterate! What happens when you change the tone, format and art form?

Step 2:

Write an outline for a fiction story that highlights the consequences of plastic pollution on the sea turtle population. Please write in a casual tone and format as paragraph. Please make this longer in length.

Of course! 🐢 Here's a **long, detailed outline** written in a **casual and story-like tone**, for a **fiction story** that highlights the **consequences of plastic pollution on sea turtles**. It's designed to sound natural and imaginative while still focusing on the environmental message.

---

#### 🐢 Story Outline: "Tide of Plastic"

The story begins on a warm, golden morning along the coast of a small tropical island. A young girl named **Mila**, who loves exploring tide pools and collecting shells, notices that the beach looks different today. Instead of shiny shells and smooth pebbles, the shore is covered with bits of colorful plastic — bottle caps, straws, old fishing lines, and even a broken sandal. She sighs and picks up a few pieces, tossing them into her beach bucket. Mila doesn't realize yet that this small act will pull her into a much bigger story — one that stretches far beyond her island.

While walking near the water, Mila spots something struggling in the shallow waves — a small **sea turtle hatchling** tangled in a plastic ring from a soda bottle. She gently frees it, names it **Kai**, and decides to help it find its way to the ocean. Kai becomes the heart of the story — a symbol of both innocence and survival. Mila's grandfather, a retired fisherman, tells her how he used to see hundreds of turtles nesting on the beach every year, but now only a few return. He shakes his head sadly, saying, "The ocean is sick, Mila. And it's because of us."

As the story unfolds, the narrative switches between Mila's world on land and Kai's journey under the sea. Through Kai's eyes, readers experience the ocean's beauty and its dangers. At first, everything seems magical — coral gardens glowing like rainbows, schools of shimmering fish, and rays gliding gracefully through the deep blue. But soon, Kai encounters floating plastic bags that look like jellyfish — his favorite food. He bites one, gets it stuck in his mouth, and becomes weak. Other young turtles swim by with bellies full of plastic, unable to dive or eat properly. The water around them grows clouded with debris, and fishing nets drift like silent traps.

Meanwhile, Mila begins to notice more dead fish washing up on shore. She and her friends start collecting plastic waste every weekend, filling bags until their hands ache. They talk to tourists who leave litter behind, and though some ignore them, others begin to listen. Mila's teacher encourages her to write about the turtles for the school newspaper, and her story spreads online. Soon, people from nearby villages come to join the cleanups. The beach slowly begins to look like it once did, though there's still much to do.

Back in the ocean, Kai barely survives after swallowing too much plastic. A marine rescue team finds him floating near the surface and brings him to a rehabilitation center. By coincidence, Mila visits the same center on a school trip — and recognizes Kai by the small scar on his shell. It's an emotional reunion. She helps feed him and learns about how hundreds of turtles end up there every year because of plastic pollution. The volunteers explain how microplastics even enter the food chain, affecting not just animals but humans too. Mila realizes the problem isn't just local — it's global.

The story builds toward hope rather than despair. After several weeks, Kai recovers and is released back into the ocean. Mila and her grandfather watch from the beach as the turtle swims into the sunset. The community has learned something important — that even small changes, like refusing plastic bags or recycling properly, can ripple outwards and make a difference. Mila's final words echo the story's message: "The ocean took care of us for so long. It's our turn to take care of it."

The ending returns briefly to Kai, now grown and gliding through clearer waters. The narrator reflects that the ocean will never be perfect again, but every clean beach, every rescued turtle, and every person who chooses to care is a step toward healing. The last line leaves readers thoughtful but inspired: *"The tide of plastic can change — if we change with it."*



Step 3: Skriv dit svar til Step 3 "What happened to the results after you changed the prompt and settings?" ind i dit document:

It went from a fact driven story that *told* about the problem, to a fiction story that *showed* the readers the problem.

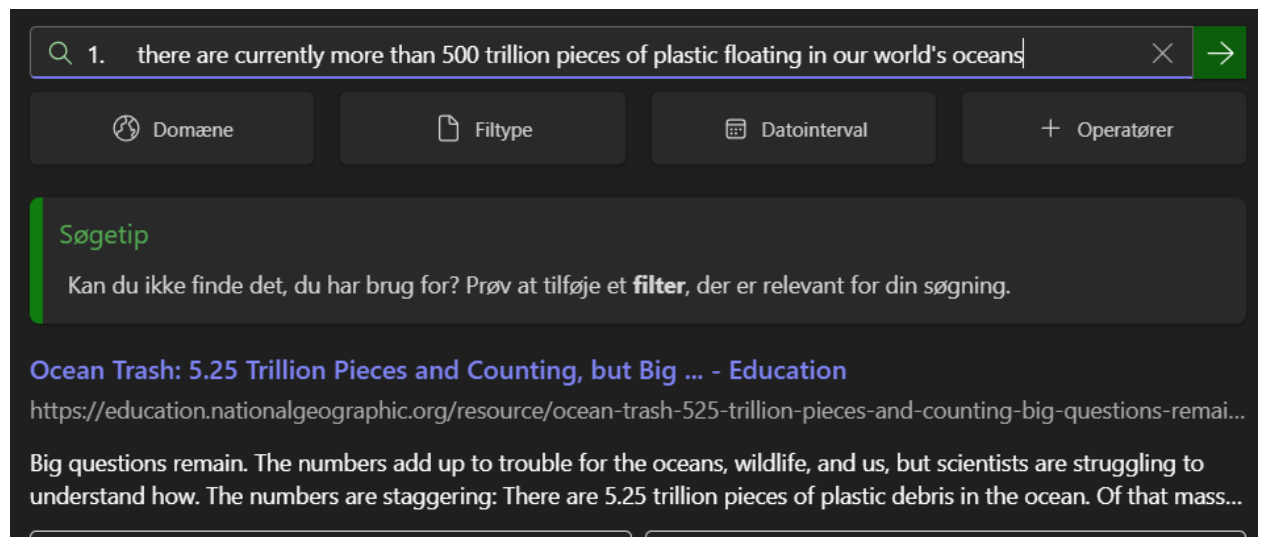
## Section 2 - Fabrications and Fact-Checking

Help Gina Fact-Check These Statements:

1. there are currently more than 500 trillion pieces of plastic floating in our world's oceans
2. One of the major threats to sea turtles worldwide is the ingestion of plastics caused by human pollution.
3. Scientists predict that by 2032 there could be more plastic than fish in the ocean.

So lets look at it:

1. According to search coach are there only 5.25 trillion pieces of plastic debris in the world



That is still a concerning number.

2. According to Search Coach plastic DOES seem to be one of five main threats to the turtles.

Q One of the major threats to sea turtles worldwide is the ingestion of plastics caused by human pollution. X →

Domæne Filtype Datointerval + Operatører

**Søgetip**

Kan du ikke finde det, du har brug for? Prøv at tilføje et **filter**, der er relevant for din søgning.

**Threats to Sea Turtles — The State of the World's Sea Turtles | SWOT**

<https://www.seaturtlestatus.org/threats-to-turtles>

All seven species of sea turtles are threatened, in large part due to human activity. There are five main threats to sea turtles: coastal development, direct take, pollution and pathogens (including plastics), climate change, and fisheries...

3. According to Search Coach it seems to be that it is first in 2050 there will be more plastic than fish in the ocean.

Q 3. Scientists predict that by 2032 there could be more plastic than fish in the ocean →

Domæne Filtype Datointerval + Operatører

**Søgetip**

Kan du ikke finde det, du har brug for? Prøv at tilføje et **filter**, der er relevant for din søgning.

**Will there be more plastic than fish in the sea? - WWF**

<https://www.wwf.org.uk/myfootprint/challenges/will-there-be-more-plastic-fish-sea>

The amount of plastic in the ocean is expected to double in the next 15 years, and it's estimated that by 2050, there could be more plastic in the ocean than fish (by weight) (Ellen MacArthur Foundation, 2016).

Step 3: which of these statements were accurate and which were not?

Conclusion: it is number 2 that turned out to be right. The rest was exaggerated.

Start med at finde 3 påstande i output fra den forrige opgave.

1. Study: a quantitative analysis showed a **50% probability of mortality** once a turtle's gut contained 14 or more pieces of plastic.
2. Another study: ~86% of green turtles examined had plastic ingestion; ~14% showed entanglement.
3. Example: more than 1,000 sea turtles estimated to die each year from entanglement alone.

Undersøg validiteten af disse 3 påstande.

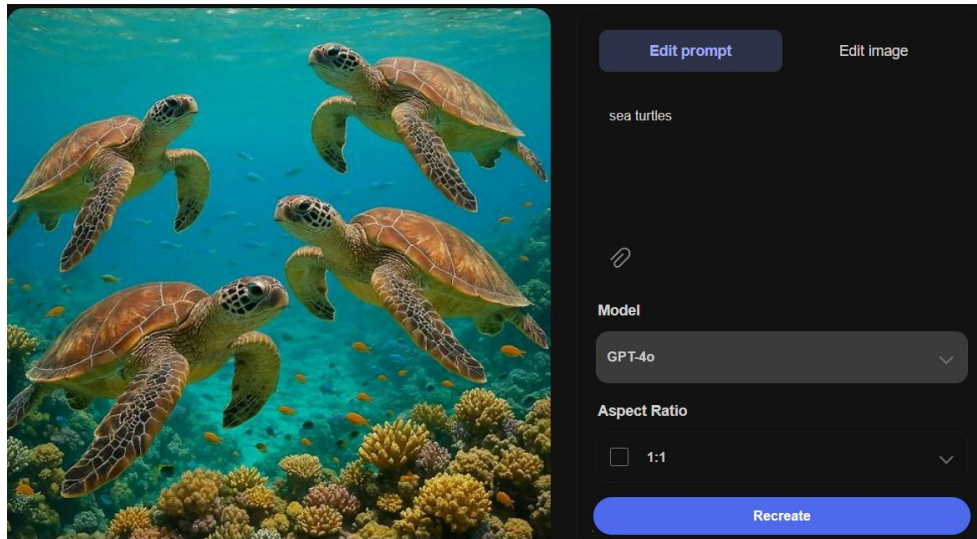
1. turns out to be true according to a analysis done by [Chris Wilcox](#) , [Melody Puckridge](#) , [Qamar A Schuyler](#) , [Kathy Townsend](#) , [Britta Denise Hardesty](#). In a paper called: **"A quantitative analysis linking sea turtle mortality and plastic debris ingestion"** found on [https://pmc.ncbi.nlm.nih.gov/articles/PMC6137038/?utm\\_source=chatgpt.com](https://pmc.ncbi.nlm.nih.gov/articles/PMC6137038/?utm_source=chatgpt.com), they state that **"We found a 50% probability of mortality once an animal had 14 pieces of plastic in its gut. "**
2. That also turned out to be true according to a study done by Yasmina Rodríguez, Frederic Vandeperre, Marco R. Santos, Laura Herrera, Hugo Parra, Ashok Deshpande, Karen A. Bjorndal , [Christopher K. Pham](#) , called "Litter ingestion and entanglement in green turtles: An analysis of two decades of stranding events in the NE Atlantic" They state in their highlights that: "Entanglement was detected in 14% of the turtles and ingestion of plastic in 86%."
3. This also turned out to be true: according to [https://www.seaturtlestatus.org/sea-turtles-plastic-pollution?utm\\_source=chatgpt.com](https://www.seaturtlestatus.org/sea-turtles-plastic-pollution?utm_source=chatgpt.com), Scientists estimate that at least [1,000 sea turtles die each year due to entanglement in plastic](#)—that's an average of more than 1 turtle every 9 hours!

**Chatgtp kom med alle kilderne til dens påstande. De viste sig alle tre at være sande. Jeg har dog tidligere oplevet falske links som Chatgtp har givet. Så det forekommer og man skal altid tjekke selv.**

# Become an AI-Powered Artist

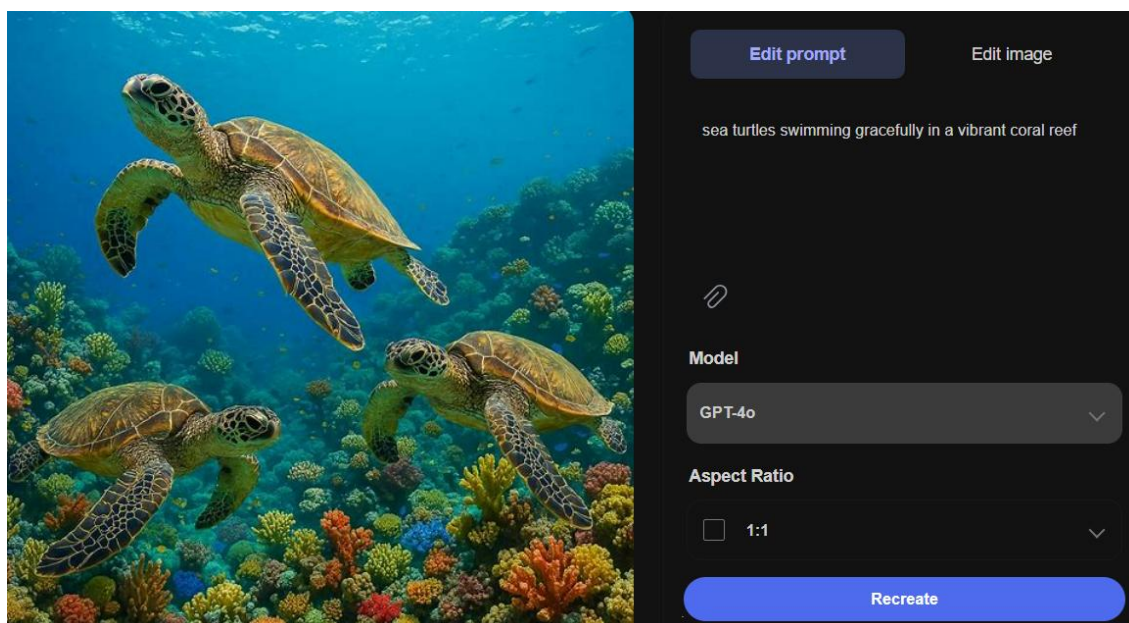
## STEP 2

Enter the prompt “sea turtles” then click *create*.



## STEP 3

Iterate your result by adding more details to your prompt. Try adding “*swimming gracefully in a vibrant coral reef*” at the end of your prompt.



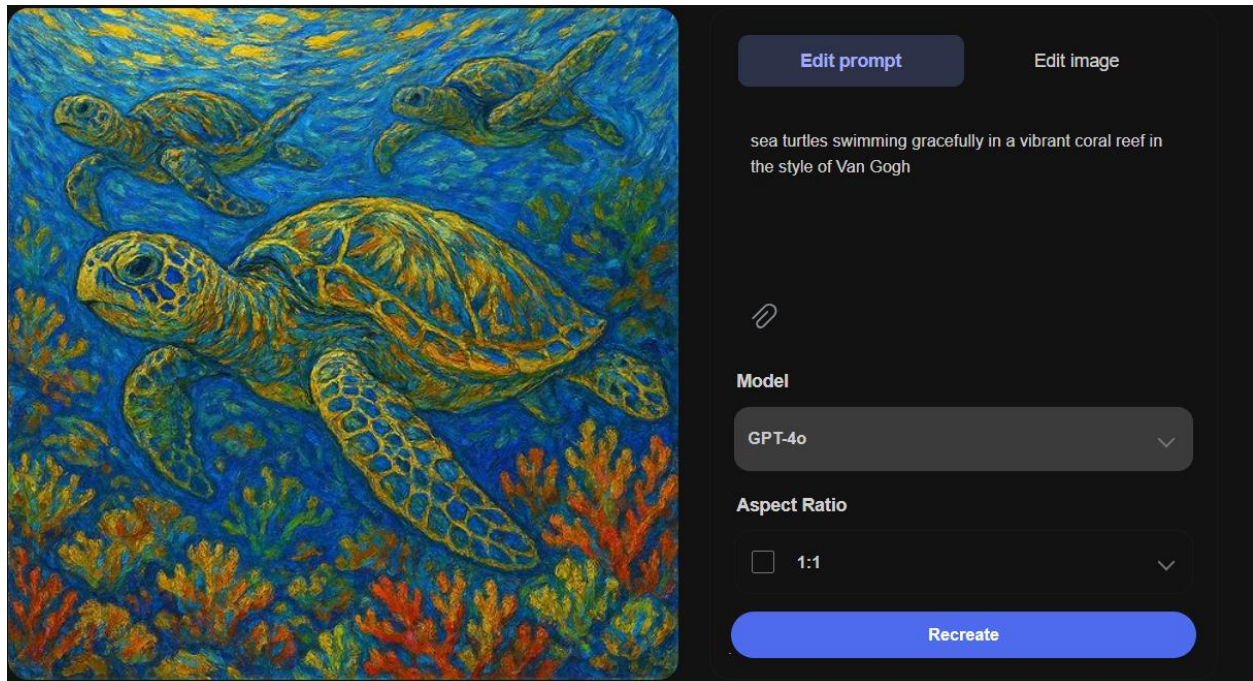
## STEP 4

Now let's stylise your output.

Add one of the following to the end of your prompt:

- a. "in photorealistic style"
- b. "in the style of Van Gogh"
- c. "in a pop art style"

I chose Van Gogh:



## Section 3 - Alex's Cosmic Adventure

### STEP 2

Fill in the blanks for this prompt:

"I want to create a character profile for a **(insert a theme)** themed online game, where this main character will be embarking on an unforgettable adventure. This character needs to have a **(insert an adjective)** personality. Please help me craft a distinct character profile, including a name, defining personality traits, a signature catchphrase, and physical attributes."

Here is my prompt:

I want to create a character profile for vampire themed online game, where this main character will be embarking on an unforgettable adventure. This character needs to have a dark personality. Please help me craft a distinct character profile, including a name, defining personality traits, a signature catchphrase, and physical attributes.

## STEP 3

Generate your character profile and share your results with the class!

**Name:**

Seraphine Veyra

**Personality Traits:**

- Mysterious, calculating, and dangerously seductive — she hides her emotions behind icy composure.
- Cruelly poetic in speech; finds beauty in tragedy and destruction.
- Suffers from a deep, unending loneliness masked by confidence and allure.
- Loyal only to her own code — despises weakness, yet secretly yearns for redemption.

**Signature Catchphrase:**

*"The moon remembers every sin we whisper to the night."*

**Physical Attributes:**

- Porcelain skin with a faint shimmer, as if kissed by frost.
- Eyes of deep amethyst that glow faintly when she uses her powers.
- Long silvery-white hair flowing like liquid light against her dark attire.
- Wears a corseted gown of black lace and leather, adorned with crimson gemstones that pulse like tiny hearts.
- Moves with an ethereal elegance, her shadow often lagging a half-second behind — a sign of her curse.

**Extra Detail (optional):**

Seraphine wields "*Lamenta*," a curved blade that drains the will to fight from her enemies. She was once a mortal poet, turned against her will, and now weaves her verses into spells that can entrance or destroy.



## STEP 4

Experiment with your results by changing the theme and adjectives to see what else the LLM can come up with.

Here is my prompt:

I want to create a character profile for a futuristic punk themed online game, where this main character will be embarking on an unforgettable adventure. This character needs to have a fun/crazy personality. Please help me craft a distinct character profile, including a name, defining personality traits, a signature catchphrase, and physical attributes.

Here is the result:

**Name:**

Nova "Glitch" Riot

**Personality Traits:**

- Wildly unpredictable and full of energy — laughs in the face of danger.
- Tech-genius hacker who lives for chaos, color, and rebellion.
- Constantly cracks jokes even during explosions — her humor hides a sharp intellect.
- Fiercely loyal to her crew, but has zero respect for authority or rules.
- Addicted to adrenaline — the louder the world, the more alive she feels.

**Signature Catchphrase:**

*"If it's not glowing, exploding, or illegal — I'm not interested!"*

**Physical Attributes:**

- Neon-pink undercut hair with fiber-optic strands that flicker to her heartbeat.
- Electric blue eyes enhanced with augmented reality overlays.
- Wears a torn leather jacket patched with holographic graffiti and glowing circuitry.
- Cybernetic arm covered in LED tattoos that shift patterns with her emotions.
- Combat boots with magnetic soles for wall-running and spark trails when she slides.

**Extra Detail (optional):**

Nova pilots a hoverboard named *"Chaos Engine"* that doubles as a hacking terminal. She's infamous across the neon slums of Neo-Tokyo for hijacking megacorp drones and turning them into street performers. Her dream? To crash the system — literally and artistically.

## Section 3 - Coding the Game

### STEP 1

Use *Microsoft Copilot* or *ChatGPT*.

### STEP 2

Enter this prompt:

“Write a HTML code for a simple ‘Guess the Number’ game.”

My prompt:

Write a HTML code for a simple ‘Guess the Number’ game.

Result:

Here's a clean, single-file version you can paste into JSFiddle (HTML pane) or save as `index.html`:

```
html Copy code

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Guess the Number</title>
<style>
  :root { font-family: system-ui, Arial, sans-serif; }
  body { margin: 2rem; display: grid; gap: 1rem; }
  .card { max-width: 560px; padding: 1rem 1.25rem; border: 1px solid #ddd; border-radius: 12px; }
  h1 { font-size: 1.4rem; margin: 0 0 .25rem; }
  p.hint { margin: 0 0 1rem; color: #555; }
  form { display: flex; gap: .5rem; align-items: center; }
  input[type="number"] { width: 7rem; padding: .5rem .6rem; font-size: 1rem; }
  button { padding: .55rem .9rem; font-size: 1rem; cursor: pointer; }
  .row { display: flex; gap: .75rem; align-items: center; flex-wrap: wrap; }
  #feedback { min-height: 1.4em; font-weight: 600; }
  .good { color: #118a36; } .bad { color: #b00020; } .info { color: #0a62b3; }
  .muted { color: #666; }
  .pill { background: #f2f2f2; border-radius: 999px; padding: .25rem .6rem; }
```



```

</style>
</head>
<body>
  <div class="card">
    <h1>Guess the Number</h1>
    <p class="hint">I am thinking of a number between <strong>1</strong> and <strong>100</strong>.
      Can you guess it in fewer than <strong>10</strong> tries?</p>

    <form id="gameForm" autocomplete="off">
      <label for="guess" class="muted">Your guess:</label>
      <input id="guess" name="guess" type="number" min="1" max="100" required />
      <button id="submitBtn" type="submit">Guess</button>
      <button id="resetBtn" type="button">Reset</button>
    </form>

    <div id="feedback" class="info" role="status" aria-live="polite"></div>

    <div class="row">
      <div class="muted">Previous guesses:</div>
      <div id="history" class="pill muted"></div>
    </div>

```

```

<div class="row">
  <div class="muted">Attempts left:</div>
  <div id="attempts" class="pill">10</div>
</div>
</div>

<script>
(function () {
  const MAX_TRIES = 10;
  let target, triesLeft, history;


  const form = document.getElementById('gameForm');
  const input = document.getElementById('guess');
  const feedback = document.getElementById('feedback');
  const attemptsEl = document.getElementById('attempts');
  const historyEl = document.getElementById('history');
  const resetBtn = document.getElementById('resetBtn');
  const submitBtn = document.getElementById('submitBtn');

  function randInt(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
  }

```

```
function startGame() {
  target = randInt(1, 100);
  triesLeft = MAX_TRIES;
  history = [];
  attemptsEl.textContent = String(triesLeft);
  historyEl.textContent = '-';
  feedback.textContent = 'Game on! Enter a number and press Guess.';
  feedback.className = 'info';
  input.disabled = false;
  submitBtn.disabled = false;
  input.value = '';
  input.focus();
}

function endGame(win) {
  input.disabled = true;
  submitBtn.disabled = true;
  if (!win) {
    feedback.innerHTML = `Out of tries! The number was <strong>${target}</strong>.`;
    feedback.className = 'bad';
  }
}
```

 Copy code

```
function addHistory(n) {
  history.push(n);
  historyEl.textContent = history.join(', ');
}

form.addEventListener('submit', function (e) {
  e.preventDefault();
  const value = Number(input.value);

  if (!Number.isInteger(value) || value < 1 || value > 100) {
    feedback.textContent = 'Please enter a whole number between 1 and 100.';
    feedback.className = 'bad';
    input.focus();
    input.select();
    return;
  }

  addHistory(value);
  triesLeft--;
  attemptsEl.textContent = String(triesLeft);

  if (value === target) {
    feedback.innerHTML = `Correct! <strong>${value}</strong> is the number. You win!`;
    feedback.className = 'good';
    endGame(true);
    return;
  }
}
```

 Copy code

```

    if (triesLeft === 0) {
      endGame(false);
      return;
    }

    if (value < target) {
      feedback.textContent = 'Too low! Try a higher number.';
      feedback.className = 'info';
    } else {
      feedback.textContent = 'Too high! Try a lower number.';
      feedback.className = 'info';
    }

    input.focus();
    input.select();
  });

  resetBtn.addEventListener('click', startGame);

  // Start on load
  startGame();

```

```

    // Enter to submit, Esc to reset (optional quality-of-life)
    input.addEventListener('keydown', (e) => {
      if (e.key === 'Escape') resetBtn.click();
    });
  })();
</script>
</body>
</html>

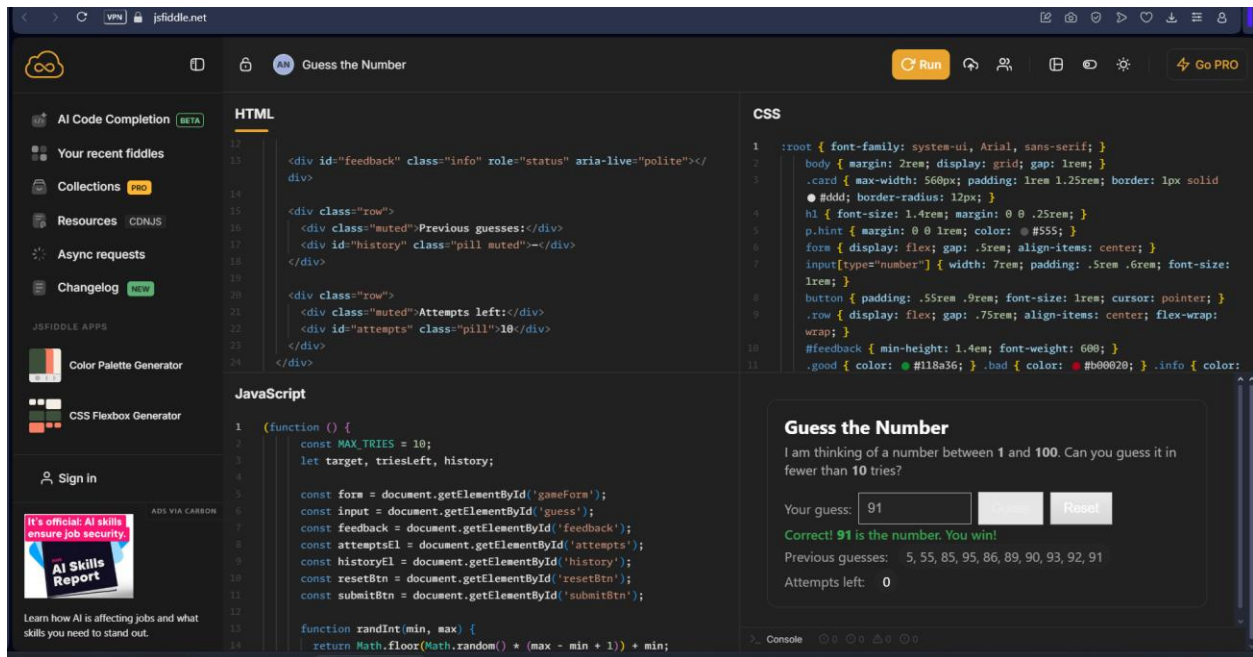
```

### STEP 3

Copy the code that was generated.

### STEP 4

Go to <https://jsfiddle.net/> and paste the code under the "HTML" box and press "RUN."



## Opgave 2 - Shortcut or Superpower

Dårlig Prompt:

kan du lave denne opgave for mig? Skriv et program, som læser en tekstlinje fra konsollen og tæller:

antallet af vokaler (a, e, i, o, u, y),

konsonanter,


cifre (0–9),

mellemrum,

og andre tegn.


Til sidst udskrives en lille rapport med tallene.

Svar fra AI:

Selvfølgelig 

Her er et færdigt og køreklar Java-program, der gør præcis det opgaven beskriver — tæller vokaler, konsonanter, cifre, mellemrum og andre tegn, og udskriver en rapport til sidst.

java

 Copy code

```
import java.util.Scanner; // importerer Scanner til input

public class CountCharactersReport { // klassens navn

    public static void main(String[] args) { // programstart
        Scanner scanner = new Scanner(System.in); // opretter scanner til brugerinput

        System.out.println("Indtast en tekstlinje:"); // beder brugeren om input
        String line = scanner.nextLine(); // læser hele linjen fra konsollen

        int vowels = 0;        // tæller vokaler
        int consonants = 0;    // tæller konsonanter
        int digits = 0;        // tæller cifre
        int spaces = 0;        // tæller mellemrum
        int others = 0;        // tæller andre tegn

        String vowelSet = "aeiouyAEIOUY"; // definerer hvilke bogstaver der er vokaler
```

```
        for (int i = 0; i < line.length(); i++) { // gennemgår hvert tegn
            char ch = line.charAt(i); // henter tegn nr. i

            if (Character.isDigit(ch)) { // tjekker om det er et tal
                digits++;
            } else if (ch == ' ') { // tjekker om det er mellemrum
                spaces++;
            } else if (Character.isLetter(ch)) { // tjekker om det er et bogstav
                if (vowelSet.indexOf(ch) >= 0) { // hvis det findes i vokal-listen
                    vowels++;
                } else { // ellers må det være en konsonant
                    consonants++;
                }
            } else { // alt andet
                others++;
            }
        }
    }
}
```

```

        // Udskriver rapporten
        System.out.println("\n=== Rapport ===");
        System.out.println("Vokaler: " + vowels);
        System.out.println("Konsonanter: " + consonants);
        System.out.println("Cifre: " + digits);
        System.out.println("Mellemrum: " + spaces);
        System.out.println("Andre tegn: " + others);
    }
}

```

God prompt:

Hvis jeg vil lave et program, som læser en tekstlinje fra konsollen og tæller:

- antallet af vokaler (a, e, i, o, u, y),
- konsonanter,
- cifre (0–9),
- mellemrum, og andre tegn.

Der til sidst udskrives en lille rapport med tallene,

Kan du lære mig hvordan man laver dette program, men uden at vise mig det endelige resultat? Kan du give en gennemgang i hvad jeg behøver at vide for at lære dette?

Hvis jeg vil lave et program, som læser en tekstlinje fra konsollen og tæller:

- antallet af vokaler (a, e, i, o, u, y),
- konsonanter,
- cifre (0–9),
- mellemrum, og andre tegn.

Der til sidst udskrives en lille rapport med tallene,

Kan du lære mig hvordan man laver dette program, men uden at vise mig det endelige resultat? Kan du give en gennemgang i hvad jeg behøver at vide for at lære dette?

Ja — det er en rigtig god måde at lære på!

Lad os tage det trin for trin, så du selv forstår, *hvordan* man bygger programmet, uden at jeg viser den færdige kode.

Du lærer samtidig en masse grundlæggende Java-tanker undervejs.

## ✖ 1. Forstå opgaven

Du skal skrive et program, der:

1. Læser tekst fra brugeren.
2. Undersøger hvert tegn ét ad gangen.
3. Tæller, hvor mange der er:
  - vokaler ( a, e, i, o, u, y )
  - konsonanter (andre bogstaver)
  - cifre (0–9)
  - mellemrum ( ' ' )
  - alt andet (tegnsætning, symboler osv.)
4. Udskriver resultatet til sidst.

Så det handler mest om **streghåndtering (Strings)** og **tegnkontrol (char)**.

## ⚙ 2. Hvad du skal vide i Java

Her er de nøgle-begreber du skal kende for at kunne bygge programmet:

### ♦ a) Hvordan man læser tekst fra brugeren

Du skal bruge en `Scanner`.

Eksempel (du behøver ikke skrive dette nu):

java

Copy code

```
Scanner input = new Scanner(System.in);  
String line = input.nextLine();
```

Det betyder:

- Opret en scanner, der lytter på tastaturet.
- Gem den tekst, brugeren skriver, i en **String-variabel**.




### ♦ b) Hvad en `String` egentlig er

En `String` er en **sekvens af tegn**.

Du kan tænke på den som en liste af bogstaver:

arduino

 Copy code

```
"Hej" → 'H' 'e' 'j'
```


For at kigge på ét tegn ad gangen, bruger man et **loop** og metoden

`line.charAt(i)` hvor `i` er et tal (position i teksten).

### ♦ c) Hvordan man går gennem hele teksten

Du bruger typisk et **for-loop**:

pgsql

 Copy code

```
for (int i = 0; i < line.length(); i++) {  
    char ch = line.charAt(i);  
    // undersøg 'ch'  
}
```

Her betyder `line.length()` "hvor mange tegn der er".


#### ♦ d) Hvordan man genkender tegn

Java har nogle smarte værktøjer i klassen `Character` :

- `Character.isLetter(ch)` → sand, hvis det er et bogstav
- `Character.isDigit(ch)` → sand, hvis det er et tal
- `Character.isWhitespace(ch)` → sand, hvis det er mellemrum, tab eller linjeskift

Derudover kan du sammenligne direkte:


java

 Copy code

```
if (ch == 'a' || ch == 'e' || ch == 'i' ...) { ... }
```

eller samle dine vokaler i en streng og søge efter dem:

java

 Copy code


```
"aeiouyAEIOUY".indexOf(ch) >= 0
```

(det betyder: findes `ch` i denne liste?)

#### ♦ e) Hvordan man tæller

Du laver fem små "tællere" – variabler, der starter på 0:


java

 Copy code

```
int vowels = 0;  
int consonants = 0;  
int digits = 0;  
int spaces = 0;  
int others = 0;
```

Hver gang du finder et tegn, øger du den relevante tæller:

java


 Copy code

```
vowels++;
```

#### ♦ f) Hvordan man udskriver resultatet

Til sidst viser du resultaterne i konsollen:

java

 Copy code

```
System.out.println("Vokaler: " + vowels);
```

(du skal gøre det for alle fem typer).

### 3. Læringsplan i 5 trin

Sådan kan du selv bygge programmet trin for trin:

#### 1. Start simpelt:

Lav et program der bare læser en tekst og viser den igen.

#### 2. Tilføj loop:

Få det til at gennemgå hvert tegn og printe det (så du kan se hvordan `charAt()` virker).

#### 3. Tæl én kategori ad gangen:

Fx start med at tælle vokaler – print resultatet.

#### 4. Udvid:

Tilføj tælling for cifre, konsonanter, mellemrum og andre tegn.

#### 5. Opsummer:

Når alle tællere virker, udskriv rapporten samlet.



## ✿ 4. Ekstra: Lav små tests

Test med sætninger som:

```
nginx
```

[Copy code](#)

```
Hej med dig 123!
```

eller

```
nginx
```

[Copy code](#)

```
AaEeIiOoUuYy
```

og tjek manuelt, om dine tællinger passer.

## ✓ 5. Hvad du bør øve dig på

- Hvordan `String` og `char` fungerer.
- Forskellen på `==` og `.equals()` (til senere brug).
- Hvordan loops og if-betingelser arbejder sammen.
- Hvordan man tæller med `int`-variabler.

## ✿ Trin 1: Læs tekst fra brugeren

Når et program skal tage input fra brugeren, skal vi bruge `Scanner`-klassen.

Den ligger i pakken `java.util`, så vi skal importere den øverst i koden.

💡 Prøv at skrive dette i din editor:

```
java
```

[Copy code](#)

```
import java.util.Scanner; // giver adgang til Scanner-klassen

public class TextReaderExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // opretter Scanner til tastatur
        System.out.println("Skriv en tekstlinje:"); // beder brugeren om input


        String line = scanner.nextLine(); // læser hele linjen som tekst
        System.out.println("Du skrev: " + line); // viser det brugeren skrev
    }
}
```

## Hvad sker der her?

1. `Scanner scanner = new Scanner(System.in);`  
→ Opretter et objekt der læser fra tastaturet.
2. `nextLine()`  
→ Venter på, at du taster noget, og trykker Enter.
3. `System.out.println(...)`  
→ Skriver tekst ud i konsollen.

👉 Prøv at køre programmet — det burde bare vise:

yaml

 Copy code

Skriv en tekstlinje:

Hej med dig

Du skrev: Hej med dig

## ✿ Trin 2: Kig på teksten ét tegn ad gangen

Nu skal du lære, hvordan man "ser" ét tegn ad gangen i teksten, ved hjælp af et **for-loop**.

Tilføj dette i stedet for udskriften i koden ovenfor:

java

Copy code

```
for (int i = 0; i < line.length(); i++) {  
    char ch = line.charAt(i); // henter tegnet på position i  
    System.out.println("Tegn #" + i + ": " + ch);  
}
```

### 🔍 Hvad sker der nu?

- `line.length()` = antallet af tegn i teksten.
- `line.charAt(i)` = henter det enkelte tegn.
- Du udskriver alle tegn ét for ét.

Eksempel:

yaml

Copy code

```
Skriv en tekstlinje:  
Hej  
Tegn #0: H  
Tegn #1: e  
Tegn #2: j
```

## ✿ Trin 3: Lav tællere og find ud af, hvad tegnene er

### 1 Opret tællervariabler

Du skal bruge én tæller til hver kategori, fx:

java

Copy code

```
int vowels = 0;    // vokaler  
int consonants = 0; // konsonanter  
int digits = 0;    // cifre  
int spaces = 0;    // mellemrum  
int others = 0;    // andre tegn
```


Alle starter på 0.

Hver gang du finder et tegn af den type, lægger du 1 til den rigtige tæller.

## 2 Undersøg hvert tegn i et loop

Du har allerede loopet:

java

 Copy code

```
for (int i = 0; i < line.length(); i++) {  
    char ch = line.charAt(i);  
}
```

Inde i dette loop skal du nu lave **betingelser**, der tjekker, hvilken slags tegn du har med at gøre.

## 3 Tjek hvilken type tegn det er

Her er nogle nyttige måder at tjekke på:

Type	Java-metode	Eksempel
Tal	<code>Character.isDigit(ch)</code>	sand hvis <code>ch</code> er 0-9
Bogstav	<code>Character.isLetter(ch)</code>	sand hvis <code>ch</code> er a-z eller A-Z
Mellemrum	<code>ch == ' '</code>	sand hvis det er et almindeligt mellemrum


Når du ved, at det *er* et bogstav, skal du afgøre, om det er en **vokal** eller **konsonant**.

#### 4 Tjek om et bogstav er en vokal

Der er to måder at gøre det på:

##### a) Den enkle (men lidt lang)


java

 Copy code

```
if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' || ch == 'y' ||  
    ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U' || ch == 'Y') {  
    vowels++;  
}
```

##### b) Den smarte (kortere)

java


 Copy code

```
String vowelSet = "aeiouyAEIOUY";  
if (vowelSet.indexOf(ch) >= 0) { // findes ch i strengen?  
    vowels++;  
}
```

#### 5 Brug `else if` for at dække alle typer

Når du sætter det sammen, bliver logikken sådan her (i pseudokode):

yaml

 Copy code

```
for hvert tegn i teksten:  
    hvis det er et ciffer → +1 til digits  
    ellers hvis det er et mellemrum → +1 til spaces  
    ellers hvis det er et bogstav:  
        hvis det er en vokal → +1 til vowels  
        ellers → +1 til consonants  
    ellers → +1 til others
```


Du kan selv prøve at skrive den kode nu (uden at kigge på færdige løsninger).



## 6 Udskriv resultatet til sidst

Når du er færdig med loopet, kan du vise resultaterne:

java


 Copy code

```
System.out.println("Vokaler: " + vowels);  
System.out.println("Konsonanter: " + consonants);  
System.out.println("Cifre: " + digits);  
System.out.println("Mellemlum: " + spaces);  
System.out.println("Andre tegn: " + others);
```

## Prøv selv:

1. Kombinér det du lavede i trin 2 med tællerne her.
2. Test med en tekst som:

nginx

 Copy code

Hej med dig 123!

3. Udskriv de fem tal.

## Trin 4: Forbedr programmet


### 1) Håndtér store og små bogstaver

Når man sammenligner bogstaver, er det smart at gøre alt til **små bogstaver** først.

Så slipper du for at teste både 'A' og 'a'.

#### Eksempel:

java

 Copy code


```
char lower = Character.toLowerCase(ch);
```

Nu kan du bare sammenligne `lower` med 'a', 'e', osv.

## ✿ 2) Tilføj understøttelse for danske vokaler (æ, ø, å)

Hvis du vil tælle æ, ø og å som vokaler, skal du bare tilføje dem i din vokalliste:

java

 Copy code

```
String vowelSet = "aeiouyæøå";
```

Husk at lave både små og store versioner, **eller** bare konvertere alle tegn til små som nævnt ovenfor.


## ⚙️ 3) Ignorer tegn, der ikke tæller som bogstaver

Nogle tegn som "—", ".", ",", osv. skal ikke regnes som konsonanter.

Derfor er det vigtigt, at du **kun** tæller konsonanter, når `Character.isLetter(ch)` er sand, og tegnet *ikke* er en vokal.

Det kan fx se sådan ud i pseudokode:

markdown

 Copy code


hvis tegnet er et bogstav:

hvis det er en vokal → +1 til vowels  
ellers → +1 til consonants

#### 4) Gør udskriften pænere (rapportform)

Når du udskriver resultater, kan du formatere det pænt, fx:


yaml

 Copy code

```
=== Tegnrapport ===  
Vokaler:      4  
Konsonanter:   6  
Cifre:         2  
Mellemrum:     1  
Andre tegn:    3  
=====
```

Dette kan du opnå med `System.out.printf()` i stedet for `println()`:

java

 Copy code


```
System.out.printf("%-15s %d%n", "Vokaler:", vowels);
```

Det gør, at tallene står pænt under hinanden.

#### 5) Tilføj totaltegn (valgfrit)

Hvis du vil gøre rapporten lidt mere interessant, kan du udregne:


java

 Copy code

```
int total = vowels + consonants + digits + spaces + others;
```

Og til sidst skrive:

java

 Copy code

```
System.out.println("Samlet antal tegn: " + total);
```

## 6) Test din forbedrede version

Brug forskellige sætninger for at se, hvordan den reagerer:

Input	Forventet fokus
Hej 123!	Skal tælle både bogstaver, tal og symbol
Åh nej...	Skal tælle æ/ø/å korrekt
AeIoUy	Skal genkende både store og små vokaler

## Opsummering af hvad du nu har lært

Du kan nu:

1. Læse tekst med `Scanner`.
2. Gennemgå den med `for` og `charAt()`.
3. Tælle forskellige tegn ved hjælp af `if` og `Character`-metoder.
4. Forbedre programmet med pæn output og ekstra funktioner.

## Trin 5: Test og Fejlsøgning

### 1) Hvorfor teste?


Når man skriver et program, der arbejder med mange små dele (tegn i dit tilfælde), kan selv en lille fejl – fx en manglende "else" – give forkerte tal.

Derfor tester man for at sikre, at logikken gør præcis det, man forventer.

## 🔍 2) Test en kategori ad gangen

Kør programmet flere gange med simple inputs, så du tester *én ting ad gangen*.

Lav fx en lille tabel over dine testcases:

Test	Input	Forventet	Fokus	
1	abc	1 vokal, 2 konsonanter	Bogstaver	
2	AEIOUY	6 vokaler	Store bogstaver	
3	12345	5 cifre	Tal	
4		4 mellemrum	Whitespace	
5	!?.,	4 andre tegn	Symboler	
6	Hej 123!	3 konsonanter, 1 vokal, 3 cifre, 1 space, 1 andet	Blandet	


Når du kører disse og sammenligner resultatet, kan du hurtigt se, hvis noget tælles forkert.

## 🔧 3) Brug print-debugging

Et simpelt trick: print, hvad der sker under loopet.

Fx før du er sikker på, at din `if`-struktur virker:


java

 Copy code

```
System.out.println("Tegn: " + ch);
```

eller

java

 Copy code

```
System.out.println("Tegn: " + ch + " → " + type);
```

Hvor `type` er en tekst, du selv skriver ind (fx "vokal", "tal", "andet").

Så kan du se programmet arbejde trin for trin.

Når alt virker, kan du fjerne de ekstra `println()`-linjer igen.

#### ⚙️ 4) Typiske fejl og hvordan du spotter dem

Problem	Forklaring	Løsning
Æ/Ø/Å bliver ikke talt som vokaler	Du har ikke medtaget dem i vokallisten.	Tilføj dem i <code>vowelSet</code> eller brug <code>toLowerCase()</code> .
Store bogstaver virker ikke	Sammenligningen er case-sensitive.	Brug <code>Character.toLowerCase(ch)</code> .
Tal bliver talt som "andre tegn"	Du har glemt <code>Character.isDigit(ch)</code> .	Tilføj <code>else if (Character.isDigit(ch))</code> .
Mellemrum bliver ignoreret	Du tjekker ikke for <code>' '</code> .	Tilføj en betingelse for space.
Tegn tælles dobbelt	Du bruger flere <code>if</code> i stedet for <code>else if</code> .	Brug <code>else if</code> – kun én betingelse må gælde pr. tegn.

#### 🧩 5) Test systematisk

Når du har rettet fejlene, test igen med flere kombinationer og edge cases ("kanter"):

- En helt tom linje (`""`)
- En linje med kun ét tegn
- En lang linje med blandede symboler
- En tekst med både danske og engelske bogstaver

Jo flere tests du laver, jo mere sikker bliver du på, at koden fungerer i alle situationer.

#### 💡 6) Tænk som en detektiv

Når noget ikke virker:

1. **Afgræns problemet** → Hvilken del af koden fejler?
2. **Print værdier** → Hvad er faktisk i `ch` lige nu?
3. **Læs fejlen højt** → Ofte ser du mønstret, når du gentager det for dig selv.
4. **Lav små ændringer** → Test én ting ad gangen.

## ✿ Trin 6: Udvid og Reflektér

Du har nu et program, der kan:

- læse tekst,
- gennemgå hvert tegn,
- klassificere det,
- og udskrive en lille rapport.

Men i den virkelige verden stopper læringen ikke dér — nu kan du begynde at **lege med ideer**, forbedre, og forstå hvorfor du gjorde, som du gjorde.

### 🚀 1) Udvid programmet med ekstra funktioner

Her er nogle oplagte idéer, du kan bygge ovenpå din kode:

#### ♦ a) Beregn procentfordeling

I stedet for bare at vise *antal*, kan du beregne, hvor stor en procentdel af teksten der er vokaler, konsonanter, osv.

Formel:

ini

📋 Copy code

```
procent = (antal * 100.0) / totalAntal
```

Output kan fx se sådan her:

makefile

📋 Copy code


```
Vokaler: 10 (25%)  
Konsonanter: 20 (50%)  
Cifre: 5 (12%)  
Andre: 5 (13%)
```

#### ♦ b) Tæl ord

Et ord er typisk alt, der er adskilt af mellemrum.

Du kan tælle dem ved at splitte teksten:

java

 Copy code

```
String[] words = line.trim().split("\\s+");
```


`words.length` giver så antal ord.

---

#### ♦ c) Find længste ord

Når du nu allerede har `words`, kan du lede efter det længste:

java

 Copy code

```
String longest = "";  
for (String w : words) {  
    if (w.length() > longest.length()) {  
        longest = w;  
    }  
}
```




#### ♦ d) Gem resultater i en fil

Hvis du vil gemme rapporten, kan du bruge `PrintWriter` eller `FileWriter`.

Fx:

java

 Copy code

```
PrintWriter out = new PrintWriter("rapport.txt");  
out.println("Vokaler: " + vowels);  
out.close();
```

Det gør, at programmet skriver sin rapport til en fil i stedet for kun at vise den på skærmen.

---

#### 🧠 2) Reflektér over, hvad du har lært

Det her er næsten den vigtigste del.

Tænk over spørgsmål som:

- Hvad skete der, da jeg kørte mit første program?
- Hvad var det sværeste at forstå — `if`-logikken, loops, eller input?
- Hvordan ville jeg forklare programmet til en klassekammerat?
- Hvad gør jeg næste gang, jeg skal tælle noget i tekst?

At kunne **forklare** sin kode er det, der adskiller en *der skriver kode* fra en *der forstår kode*.

### ✿ 3) Næste skridt (bonus-idéer)

Når du føler dig sikker i dette emne, kan du prøve:

- 📊 **Tælle bogstaver individuelt** (hvor mange a'er, b'er, c'er ...)
- 📄 **Analysere flere linjer tekst** (ikke kun én linje)
- 📋 **Lave et mini-tekstværktøj** der viser: antal ord, antal bogstaver, længste ord, gennemsnitlig ordlængde
- 💡 **Brug GUI** (grafisk interface) i stedet for konsol, fx med JavaFX eller Swing

### 🎲 4) Konklusion

Du har nu gennemført hele læringsforløbet:

Trin	Du har lært at...	📄
1	Læse tekst med <code>Scanner</code>	
2	Gå gennem tekst med <code>for</code> og <code>charAt()</code>	
3	Tælle tegn via <code>if</code> -betingelser	
4	Forfine og håndtere flere sprog/tegn	
5	Teste og fejlsøge systematisk	
6	Udvide og reflektere som en programmør	

👏 Du har med andre ord **bygget, forstået og forbedret** et lille tekstanalyseprogram fra bunden — præcis som man gør i softwareudvikling.

## Opgave 3 – Debugging

```

C:\Users\pml\Idea\workspace\25H2\25H2\src>javac Main.java
This is the correct list:
[None, None, None]

Please enter the name of the person you want to remove:
>
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index -1 out of bounds for length 3
    at java.base/java.util.Objects.checkIndex(Objects.java:365)
    at java.base/java.util.ArrayList.remove(ArrayList.java:552)
    at Main.main(Main.java:27)

Process finished with exit code 1

```



**Main (1).java**  
File

jeg fik den her fil af min lærer: når man kører den og skriver et navn ind, så kommer den med denne fejl: `Exception in thread "main" java.lang.IndexOutOfBoundsException: Index -1 out of bounds for length 3`

at  
`java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:100)`  
 at  
`java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:106)`  
 at  
`java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:302)`  
 at `java.base/java.util.Objects.checkIndex(Objects.java:365)`  
 at `java.base/java.util.ArrayList.remove(ArrayList.java:552)`  
 at `Main.main(Main.java:27)`

Process finished with exit code 1

jeg bruger selv IntelliJ IDEA 2025.2.1 (Ultimate Edition), Build #IU-252.25557.131, built on August 27, 2025 og sidder på en windows 11 Home version 25H2. kan du hjælpe med at finde ud af hvordan jeg får den til at virke igen?




**Svar fra AI:**

Thought for 17s

Ja—det virker faktisk som forventet: din kode crasher, fordi den prøver at fjerne et element på indeks -1.

## Hvad sker der?

java

 Copy code


```
int indexOfPersonToRemove = persons.indexOf(new Person(name));
persons.remove(indexOfPersonToRemove); // BOOM hvis indexOf == -1
```

`indexOf(...)` bruger `equals(...)` til at finde elementet. Din `Person`-klasse **overrider ikke** `equals` / `hashCode`, så et nyt `Person(name)`-objekt er ikke "lig" det, der ligger i listen—selv om navnet matcher. Derfor får du -1, og `remove(-1)` giver `IndexOutOfBoundsException`.

## Løsning A (klassisk): sammenlign personer på navn

Overstyr `equals` og `hashCode` i `Person` (jeg gør det case-insensitive) og håndtér "ikke fundet" før du fjerner:

java

 Copy code

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        ArrayList<Person> persons = new ArrayList<>();
        persons.add(new Person("Anna"));
        persons.add(new Person("Bob"));
        persons.add(new Person("Carl"));
    }
}
```

```
Scanner keyboard = new Scanner(System.in);

System.out.println("This is the current list:");
System.out.println(persons);
System.out.println("-----");
System.out.print("Please enter the name of the person you would like to remove: ");

String name = keyboard.nextLine().trim();

int index = persons.indexOf(new Person(name)); // virker nu pga. equals
if (index >= 0) {
    Person removed = persons.remove(index);
    System.out.println("Removed: " + removed);
} else {
    System.out.println("No person with name '" + name + "' found.");
}

System.out.println("This is the current list: " + persons);
}
}

class Person {
    private final String name;
    public Person(String name) { this.name = name; }
    public String getName() { return name; }
    @Override public String toString() { return name; }

    @Override public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Person)) return false;
        Person other = (Person) o;
        return name != null && other.name != null &&
            name.equalsIgnoreCase(other.name);
    }
}
```

```
@Override public int hashCode() {  
    return name == null ? 0 : name.toLowerCase().hashCode();  
}  
}
```

[Copy code](#)

## Løsning B (nem og tydelig): fjern ved at matche navn direkte

Så behøver du ikke røre `equals/hashCode` :

```
java  
  
String name = keyboard.nextLine().trim();  
boolean removed = persons.removeIf(p -> p.getName().equalsIgnoreCase(name));  
  
if (removed) {  
    System.out.println("Removed: " + name);  
} else {  
    System.out.println("No person with name '" + name + "' found.");  
}  
System.out.println("This is the current list: " + persons);
```

[Copy code](#)

Husk i så fald at have en `getName()` i `Person` .

## Bonus – undgå crashet uanset hvad

Selv hvis du fortsætter med `indexOf` , tjek altid for `-1` før `remove` :

```
java  
  
int idx = persons.indexOf(new Person(name));  
if (idx >= 0) persons.remove(idx);  
else System.out.println("Not found.");
```

[Copy code](#)

Hvis du stadig får problemer i IntelliJ:

- Sørg for at filnavn = klassens navn (`Main.java` ↔ `public class Main` ).
- Slet evt. dubletter som `Main (1).java` .
- Tjek Run Configuration → *Main class*: `Main` .
- Build → Rebuild Project.

Sig til, hvilken løsning du vil bruge (A eller B), så tilpasser jeg koden helt præcist til din fil.

## Opgave 4 - Måder at bruge Generativ AI på

I artiklen [Practices for the Application of Generative AI in Programming](#)

[Education](#) opremses mange forskellige måder hvorpå Generativ AI kan bruges til at understøtte læring, specifikt indenfor programmering.

Du har i løbet af de forrige opgaver allerede været igennem et par af punkterne.

Udvælg ét af punkterne, og lav et eksempel hvor du bruger Generativ AI på den udvalgte måde.

### Skriv hvilket punkt du har valgt ind i dit dokument.

#### Practice: Concept teaching/explanation

Instead of asking a human instructor, students can ask the tool to explain a programming concept. This could include the generation of some example code which implements/applies the concept. Students can make this teaching/explanation more context-specific, e.g. by providing some extra

information in the prompt like the task they are working on or elements of the concept they do not understand yet.

To refine the level of teaching/explanation, this practice could be combined with a persona prompt, for example: "From now on, act as computer science teacher teaching programming in xxx at bachelor level." (White et al., 2023)

Examples: "What's the difference between checked and unchecked exceptions in Java? Give code examples for each."

Source: (Lau & Guo, 2023)

### My prompt:

Hey, I am having trouble understanding the whole concept of arrays, can you help me, by explaining it to me, like I was a student in 3rd grade?

Absolutely! 🧠 ✨

Let's make **arrays** super easy to understand — like something you could explain to your little brother or sister!

### 📦 Imagine a row of boxes

Think of an **array** as a **row of boxes** sitting side by side.

Each box can **hold one item** — like a number, a word, or something else.

Example:

You have 5 boxes to store your favorite numbers.

css

📄 Copy code

```
[ 7 ][ 2 ][ 9 ][ 4 ][ 6 ]
```

That's an **array** with 5 spots.

### 🏷️ Each box has an address

Each box has a little **number label** so you can find it.

These labels start from **0**, not 1 (this is important!).

Box Label	0	1	2	3	4
What's inside	7	2	9	4	6

So:

- Box **0** holds **7**
- Box **1** holds **2**
- Box **2** holds **9**


If you want to get the number 9, you say:

👉 "Hey computer, give me the thing in box number 2!"



In code, that's written like:

java


 Copy code

```
numbers[2];
```

### You can change what's inside

Let's say you want to change the number in box 3 from 4 to 10:


java

 Copy code

```
numbers[3] = 10;
```

Now your boxes look like:

css


 Copy code

```
[ 7 ][ 2 ][ 9 ][ 10 ][ 6 ]
```

### You can loop through them

If you want to look at *every* box, you can use a **loop**, like:

java

 Copy code

```
for (int i = 0; i < 5; i++) {  
    System.out.println(numbers[i]);  
}
```


That means:

“Start at box 0, go until box 4, and print what's inside each box!”

## Why do we use arrays?

Because they help us keep many things in one place  
— instead of having lots of separate variables like:


java

 Copy code


```
int num1 = 7;  
int num2 = 2;  
int num3 = 9;  
int num4 = 10;  
int num5 = 6;
```

we can just write:

java

 Copy code

```
int[] numbers = {7, 2, 9, 10, 6};
```

 One neat line instead of five!

## Summary

Idea	Explanation
Array	A row of boxes that store many things of the same kind
Index	The box number (starts at 0)
Access	<code>array[index]</code> gets or changes what's inside
Use	Great for lists — numbers, names, scores, etc.

