

# Build a HuggingFace Agent

...

# Intro

What are HuggingFace transformers?



- Library for state-of-the-art Machine Learning for Pytorch, TensorFlow and JAX.
- Provides thousands of pretrained models to perform tasks on different modalities such as text, vision, and audio.

Source: <https://huggingface.co/docs/hub/en/transformers>

# Agenda

1. What are agents and tools
2. How to setup and initialize the agent
3. How to use predefined tools: translation, image captioning, text-to-speech
4. What other predefined tools exist
5. How to create a custom tool

# What are agents

- Large Language Model (LLM), and we're prompting it so that it has access to a specific set of tools.
- LLM generates a small piece of text

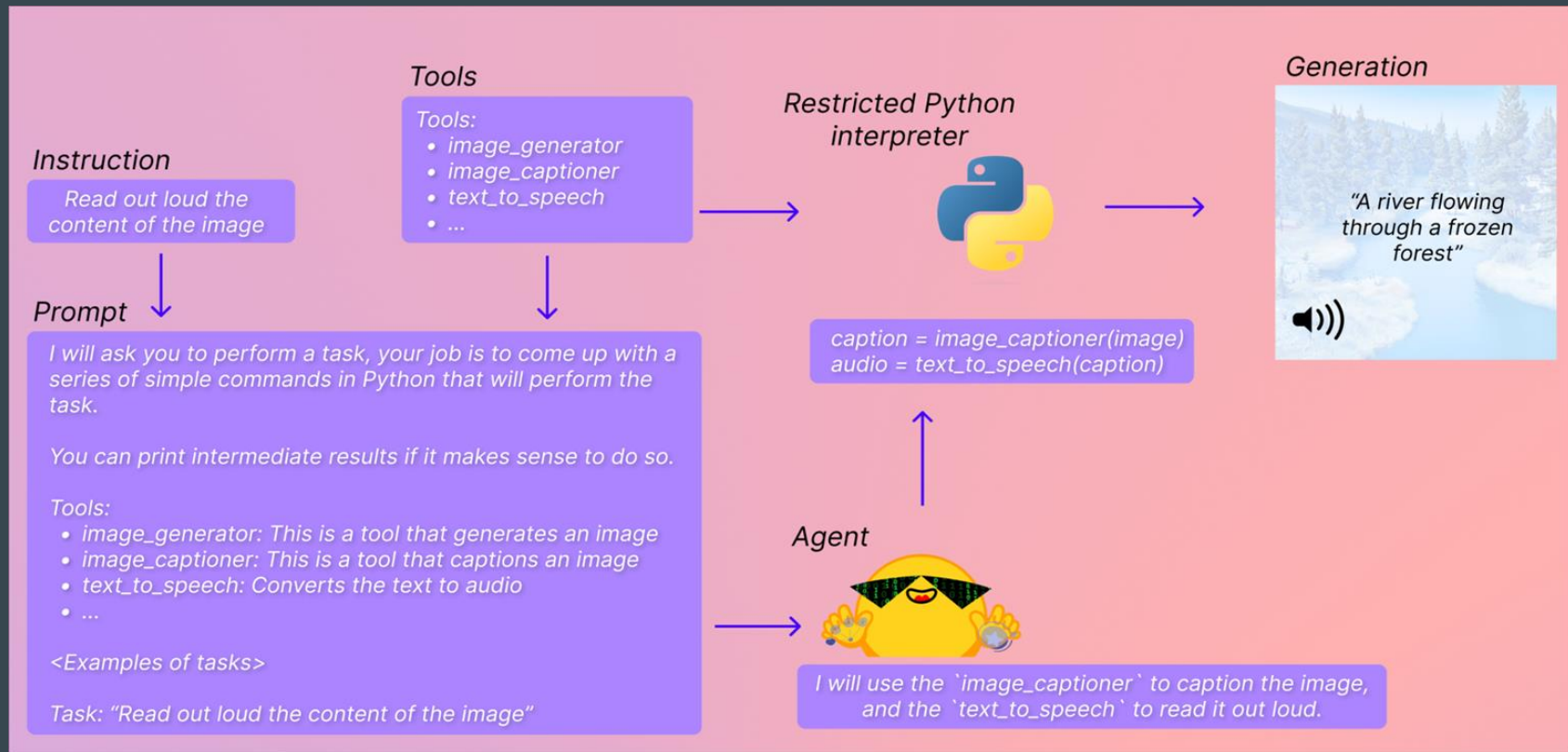


# What are tools

- Simple
- Single function with a name and a description.
- Each tool == one very simple task.



# Agents and tools



# Why should we care?

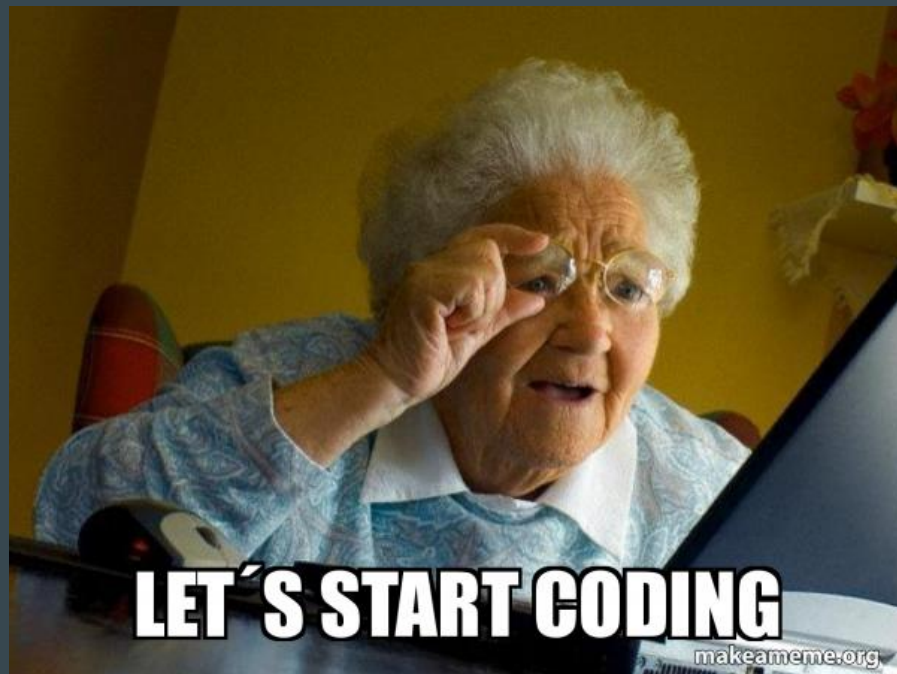
- Great interaction experience
- Instrument to pass/chain output
- Flexibility



# Hands-on: setup and initialize the agent

We need:

- HuggingFace token
- OpenAI API key
- a bit of code





# Predefined tools: Image capturing tool

Describe any image

```
agent.run("Caption the following image", image=image)
```

Input

Output



A beaver is swimming in the water

Source: [https://huggingface.co/docs/transformers/main/transformers\\_agents](https://huggingface.co/docs/transformers/main/transformers_agents)

# Translation and audio generation tool

Translate to/from over 80 languages

Voice the text easily

```
agent.run(f"Translate 'text' to Spanish and read it loud", text=description)
```

```
==Explanation from the agent==
```

```
I will use the following tools: `translator` to translate the text, then `text_reader` to read it out loud.
```

```
==Code generated by the agent==
```

```
translated_text = translator(text=text, src_lang="English", tgt_lang="Spanish")
```

```
print(f"Translated text: {translated_text}")
```

```
audio_translated_text = text_reader(translated_text)
```

```
==Result==
```

```
Translated text: una mujer con el pelo largo y una camisa de color liso
```

▶ 0:04 / 0:04 ———— 🔊 ⋮

# What other predefined tools exist (based on transformers)

- Document question answering
- Text question answering
- Image question answering
- Image segmentation
- Speech to text
- Zero-shot text classification
- Text summarization

# What other predefined tools exist (transformers-agnostic)

- Text downloader
- Text to image
- Image transformation
- Text to video

```
agent.chat("Generate a picture of rivers and lakes")
```



```
agent.chat("Transform the picture so that there is a rock in there")
```



Source: [https://huggingface.co/docs/transformers/main/transformers\\_agents](https://huggingface.co/docs/transformers/main/transformers_agents)

# Custom tool creation

Simple way to extend an agent

Possibility to push custom tool to the hub

# Conclusion

- Still experimental but promising
- Easy to start with
- Easy extendable
- Allows building input-output chains and make really smart agent
- Beware an unexpected output



# Further ideas

- Check agents of LangChain and Bedrock
- Experiment with prompting
- Custom model can be an agent



LangChain: <https://www.langchain.com/>

Amazon Bedrock: <https://aws.amazon.com/bedrock/>

**You can find code and links:**

**[https://github.com/dashapetr/hf\\_agents](https://github.com/dashapetr/hf_agents)**





**Let's stay in touch:**

**Darya Petrashka**

