

Scaling Real-World Programming with ChatGPT

Morten Heine Sørensen


mhs@formalit.dk

www.lambdaprompt.com

Scoped Carbon Accounting


2024-08-15

Is 2023 a prime?


 **Morten Heine Sørensen** • You
Not open to work, but if your project is in trouble, I may be abl...
7mo • 🌐

Math capabilities improved, but still some way to go.


M Is 2023 prime?

 No, 2023 is not a prime number.


M Why not?

 A prime number is a positive integer greater than 1 that has no positive integer divisors other than 1 and itself. 2023 can be divided evenly by 3 and 673, making it a composite number and not a prime number.


M What is 3×673 ?

 $3 \times 673 = 2019$.


M Does 3 and 673 divide 2023?


 No, $3 \times 673 = 2019$, not 2023. So 3 and 673 do not divide 2023.

M Is 2023 prime after all then?

 No, 2023 is still not a prime number.


M Why not?


 2023 can be divided evenly by 17 and so it is not a prime number. A prime number is defined as a positive integer greater than 1 that has no positive integer divisors other than 1 and itself.

 **Morten Heine Sørensen** • You
Not open to work, but if your project is in trouble, I may be abl...
6mo • 🌐

ChatGPT, I have some good news for you.


MH IS 2023 prime?

 I'm sorry, but I cannot predict whether 2023 is a prime number or not, as it is a well-known mathematical problem that has not yet been solved. However, I can tell you that 2023 is not divisible by 2 or 3, which are the first two prime numbers, but that does not necessarily mean that it is prime. To determine if 2023 is prime or not, you would need to check if it is only divisible by 1 and itself, or if it has any other divisors.

 **Morten Heine Sørensen** • You
Not open to work, but if your project is in trouble, I may be abl...
6mo • 🌐

ChatGPT are you doing coin flip to produce your answers? Yesterday it was an open mathematical problem.

MH Is 2023 a prime?

 Yes, 2023 is a prime number.

Coding with ChatGPT

M What is the typical error when a pygame is not showing screen?



A common error when a Pygame screen is not showing is that the Pygame display has not been initialized. This can be fixed by adding the following line of code before creating the display window:

```
pygame.init()
```

Another possible error is that the display update function, `pygame.display.update()` or `pygame.display.flip()`, is not being called, so the screen is not being refreshed.

Additionally, check that the surface created is not None, and the surface is properly blitted to the screen before updating the display.

Also, make sure the main loop is running and event handling is done correctly.

Lastly, it is important to close the window after the game is finished using `pygame.quit()` and `sys.exit()`

Here is the same game example but using both `pygame.display.update()` and `pygame.display.flip()`:

```
python
import pygame

# Initialize pygame
pygame.init()

# Set the size of the window
size = (700, 800)
screen = pygame.display.set_mode(size)

# Set the title of the window
pygame.display.set_caption("Example")

# Run the game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # Clear the screen
    screen.fill((255, 255, 255))

    # Draw a rectangle on the screen
    pygame.draw.rect(screen, (0, 0, 255), (50, 50, 100, 100))

    # Update a portion of the screen
    pygame.display.update((50, 50, 100, 100))
    # Update the entire screen
    pygame.display.flip()

# Exit pygame
pygame.quit()
```

In this example, the `update()` function is used to update a specific portion of the screen, which in this case is the rectangle that is drawn. And `flip()` function is used to update the entire screen.

`update()` takes a rectangle as its argument, which represents the area of the screen that needs to be updated. This can be useful if you only need to update a small portion of the screen, such as when a sprite moves, instead of updating the entire screen.

while `flip()` updates the entire screen.

It is also worth noting that `flip()` is generally considered to be the recommended method for updating the screen in pygame, as it is more efficient and works well on all platforms.

Conclusions, February 2023

- ChatGPT does not do calculations, ChatGPT tells STORIES
- ChatGPT does not execute code , ChatGPT tells STORIES
- ChatGPT does not do symbolic reasoning , ChatGPT tells STORIES
- ChatGPT makes factual errors
- ChatGPT gives inconsistent answers
- ChatGPT tells STORIES
- But: ChatGPT can help fixing bugs **faster than Google**

Scaling Real-world application development

LESSON 1: ChatGPT can provide tutorials on how to accomplish tasks that are tailored to your specific application, making them very easy and fast to use.

Full-stack representative application:

- UI built with React.
- API built on Node.js.
- Data layer with Postgres database.

Approach:

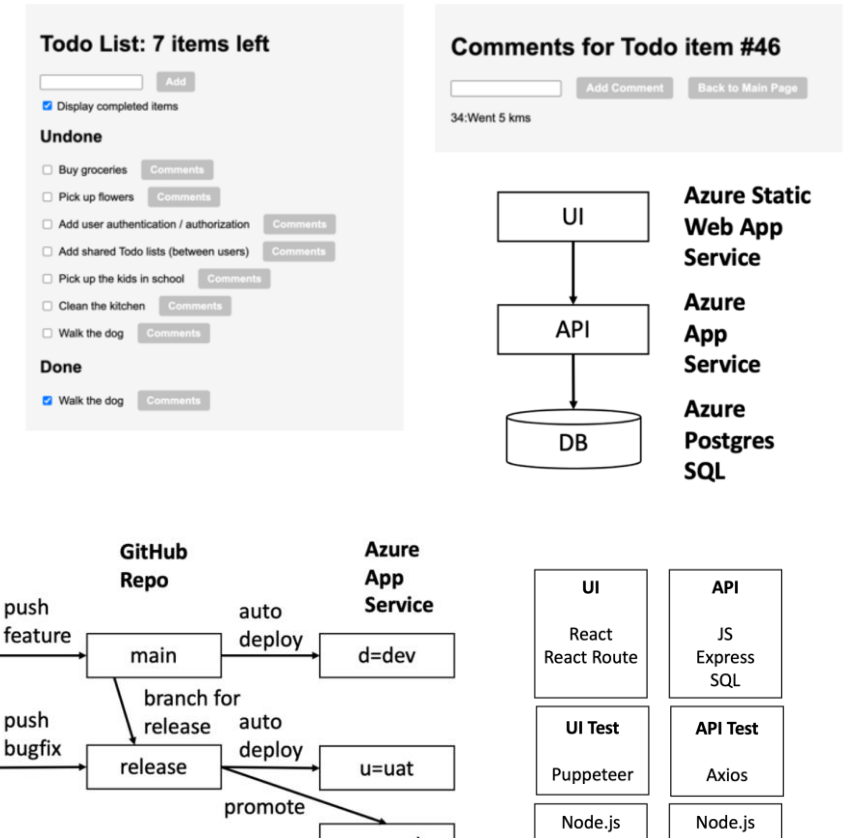
LESSON 3: Split the development into sprints and user stories, like you would with a team of developers. Then proceed sprint by sprint, user story by user story, for the same reasons as you normally do.

- Standard approach with developers
 - Sprints
 - Releases
 - Environments
 - Pull-requests
 - User-stories
 - UI Specs
 - DB and API specs
- Delegate user stories/specs to ChatGPT
- Receive pull-requests from ChatGPT

Steps:

1. Set up the project
2. Create the data model
3. Create the API
4. Create the UI
5. Create the styling
6. Create the tests
7. Refactor common parts
8. Mature the application
9. Set up environments+pipelines
10. Scale with formalism
11. Scale with developers

LESSON 5: Except for mock-ups, UI experimentation and hobby projects, most sprints should start by implementing the data model of the feature.



Dialogues – Set up project

- Install node
- Install IDE
- npx create-react-app
 - Project
 - File structure
 - Skeleton
- Run the sample application

LESSON 2: Spend the time needed to set up the local development environment to efficiently support the iterations with ChatGPT. It does not matter if this take a bit more time than ideal since it will be a one-time cost.

Dialogues – First part of UI

Please generate a React application that can maintain a list of todo items.

There should be a single page implemented in App.js.

The page should contain an input field where a text can be entered and a button called Add which creates the item. The input field and button should be in a component TodoCreator.js

The page should also contain a list of all the generated items in a component TodoRow.js

The items should be stored in local storage of the browser. The list is initially empty.

Please indicate name and contents of each file

3. 'App.js'

```
jsx Copy code

import React from 'react';
import TodoCreator from './TodoCreator';
import TodoRow from './TodoRow';

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      todos: JSON.parse(localStorage.getItem('todos')) || []
    };
  }

  addTodo = (todo) => {
    this.setState(prevState => {
      return {todos: [...prevState.todos, todo]};
    }, () => localStorage.setItem('todos', JSON.stringify(this.state.todos)))
  }

  render() {
    return (
      <div className="App">
        <TodoCreator addTodo={this.addTodo} />
        <TodoRow todos={this.state.todos} />
      </div>
    );
  }
}

export default App;
```

4. 'TodoCreator.js'

```
jsx Copy code

import React, { Component } from 'react';

class TodoCreator extends Component {
  constructor(props) {
    super(props);
    this.state = {
      input: ''
    };
  }

  handleChange = (event) => {
    this.setState({ input: event.target.value });
  }

  handleSubmit = (event) => {
    event.preventDefault();
    this.props.addTodo(this.state.input);
    this.setState({ input: '' });
  }

  render() {
    return (
      <form
        onSubmit={this.handleSubmit}
        <input type="text" value={this.state.input} onChange={this.handleChange} />
        <button type="submit">Add</button>
      </form>
    );
  }
}

export default TodoCreator;
```

LESSON 4: In Sprint 1, establish the fundamental architecture of the application, for instance a UI layer, an API layer, and the DB layer developed well enough to cover a small feature. The fundamental architecture should not only align to functional and non-functional requirements, but also to the team size and team experience.

LESSON 8: When working with ChatGPT on User Interface, ask first for a simple version. Then add remaining controls one by one or in small groups.

Okay, we'll need a few files to build this application:

1. **'package.json'**: This file keeps track of all dependencies and scripts for the project.
2. **'index.js'**: This file will serve as the entry point for our application.
3. **'App.js'**: This is the main component of our application which will render the TodoCreator and TodoRow components.
4. **'TodoCreator.js'**: This component is responsible for creating new todos.
5. **'TodoRow.js'**: This component is responsible for rendering all todo items.

5. 'TodoRow.js'

```
jsx Copy code

import React from 'react';

const TodoRow = ({ todos }) => {
  return (
    <ul>
      {todos.map((todo, index) => (
        <li key={index}>{todo}</li>
      ))}
    </ul>
  );
}

export default TodoRow;
```

To start this application, make sure you have Node.js and npm installed. Put all the above files in the same directory and run **'npm install'** to install dependencies. Use the **'npm start'** command to start the application.

LESSON 6: ChatGPT's answers should be evaluated like they come from another (sometimes less experienced) person, rather than from a flawless machine. The human developer who integrates ChatGPT's responses into the application remains liable.

LESSON 10: When starting the interaction with ChatGPT, have a dummy application working.

Dialogues – Create the server

I would like to add a Node server implemented by a file `Server.js` offering a rest API for todo items. It should offer endpoints to add an item, to mark an item done and to mark an item undone. The application just keeps the todos in memory.

In `package.json` there should be a separate script to start the server. It should listen to port 3500.

So far the React application should not use the API, so we only add `Server.js` and `package.json`, we do not alter existing react files.

Please indicate name and full contents of files that need change.

LESSON 7: When starting the API, get a simple server running with a simple example and make sure you are happy with it to some level of maturity. There may be some production hardening missing that can be covered latter, but the basic structure should be correct and satisfactory.

Let's create a simple Node server using Express that stores todos in memory and provides the specified API endpoints.

`Server.js`:

```
javascript Copy code

const express = require('express');
const bodyParser = require('body-parser');

const app = express();
app.use(bodyParser.json());

let todos = []; // Storing todos in memory

// Endpoint to add a todo
app.post('/todos', (req, res) => {
  const newTodo = req.body;
  todos.push(newTodo);
  res.json(newTodo);
});

// Endpoint to mark a todo as done
app.put('/todos/:id/done', (req, res) => {
  const id = req.params.id;
  const todo = todos.find(t => t.id === id);
  if (todo) {
    todo.done = true;
    res.json(todo);
  } else {
    res.status(404).json({ error: 'Todo not found' });
  }
});

// Endpoint to mark a todo as not done
app.put('/todos/:id/undone', (req, res) => {
  const id = req.params.id;
  const todo = todos.find(t => t.id === id);
  if (todo) {
    todo.done = false;
    res.json(todo);
  } else {
    res.status(404).json({ error: 'Todo not found' });
  }
});

app.listen(3500, () => {
  console.log('Server is running on port 3500');
});
```


Dialogues – Creating the database

I have a postgres database running locally on port 5432. Name of database is todos. User is "postgres", password is "password". Please provide the following:

1. SQL for creating a table that can hold todo items. It should include proper indices.
2. Updated version of Server.js that calls the database to insert items and fetch items

Sure! Let's go step by step:

1. Here is the SQL to create a `todos` table:

```
sql
CREATE TABLE todos (
  id SERIAL PRIMARY KEY,
  name TEXT NOT NULL,
  done BOOLEAN NOT NULL DEFAULT false
);

CREATE INDEX idx_todos_done ON todos(done);
```

This will create a `todos` table with an `id` column as the primary key, a `name` column for the todo text, and a `done` column to store the status of each todo. The `idx_todos_done` index will improve the performance of queries that filter by the `done` column.

1. We'll need to install `pg` npm package to interact with Postgres database in Node.js. Run `npm install pg` in your terminal.

1. We'll need to install `pg` npm package to interact with Postgres database in Node.js. Run `npm install pg` in your terminal.

Then here is the updated `Server.js`:

```
javascript
const express = require('express');
const { Pool } = require('pg');
const app = express();
const port = 3500;

const pool = new Pool({
  host: 'localhost',
  port: 5432,
  user: 'postgres',
  password: 'password',
  database: 'todos'
});

app.use(express.json());

app.get('/todos/done', async (req, res) => {
  const result = await pool.query('SELECT * FROM todos WHERE done = true');
  res.json(result.rows);
});

app.get('/todos/undone', async (req, res) => {
  const result = await pool.query('SELECT * FROM todos WHERE done = false');
  res.json(result.rows);
});
```

LESSON 14: Be as precise as you can in stating what you need from ChatGPT. Whenever you omit details, ChatGPT may do something else than you expect.

LESSON 15: If ChatGPT is lacking context, he may make assumptions instead of asking for clarifications. You can circumvent this by explicitly directing him initially to ask clarifications.

Dialogues – Other parts

- Create the styling
- Create the tests
- Refactor out common parts
- Mature the application
- Set up the prod environment
- Scale with formalism
- Scale with developers
- Dealing with limitations

LESSON 16: Sometimes, several iterations do not bring you closer to a solution. In these cases, consider reverting to classical techniques, like Googling, Stack Overflow, YouTube demos, reading the documentation, etc.

LESSON 17: ChatGPT currently does not know about knowledge published after September 2021

LESSON 18: ChatGPT generates several errors. It is valuable to have a slim UI test and API test that can be run locally before merging pull requests.

LESSON 20: As you review the PRs from ChatGPT, keep an eye on parts of the code that could or should be refactored. Either do the refactoring yourself or ask ChatGPT to do it.

LESSON 23: See the code by ChatGPT as your code. You must be able to account for it. That way it has same value as code developed by you or your developers.

LESSON 24: The code that you get from ChatGPT may not be ready for production, but he can help you understand what needs to be checked and how, and which changes are needed as a result of the checks.

LESSON 25: Formalize your communication with ChatGPT with initial training. That way the communication can hopefully be way more efficient and less ambiguous.

What's New?

- **Summer 2024 – Improvements and Criticism**

- Features:
 - Upload and download files, cool
 - Able to process images, cool
 - Generate HTML/CSS from image, multiple iterations needed and can only get to a certain point before it gets worse
 - Claude (see below) comparable
- UI offerings
 - Generate React from Figma (multiple) – I don't want to create Figma's
 - Fronty: create website – I want pages
 - Image2codeai – I want free trial
 - Other offerings not working – I want something working
- Customized "GPTs"
- Competitor Claude
- *Business standard*: ChatGPT costs \$700,000 daily, maker may go bankrupt in 2024
- *Hicks et al.*: ChatGPT is bullshit
- Critics claim they were right
- Proponents claim they were right
- Limitations on what code can be submitted due to privacy considerations
- McKinsey: "Inaccuracy: The most recognized and experienced risk of gen AI use." Not a problem for us.