

Web Programming

Day 7 AM

*"Life's a dance you learn as you go.
Sometimes you lead,
sometimes you follow."*

John Michael Montgomery

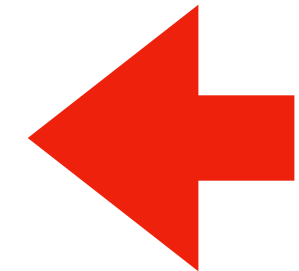
Storybook (initial)

Drehbuch, Intro, Functions
Scientific foundations
Algebraic Data Types, Snake
Applied Science

Scripting, PWA, Plotter, Excel
Objects
Classes
JS Types, JsDoc

Async Programming
Modules
Data Flow, Excel improved
Iterator Protocol, Sequences

Programming Moves
User Interfaces
Professional JS with Kolibri
Crazy JS



Personal Toolbox

For up to 10 extra points.

Submission mail to me until
06.02.26 (inclusive)
with link to repo or equivalent

Today: Moves

Programming is an activity

What is the nature of this activity?

What are the parts?

Compare with Dancing

You must learn the moves.

Then you can combine the moves and adapt to the situation at hand.

Recognize Moves

Become aware what you do.

We program collaboratively and
look for moves.

Moves - Your Choice

What counts is your moves, not mine.

There are signal - reaction schemes

Try to have multiple options

Moves

- 0: Explore
- 1: Goal Card
- 2: Start at the End
- 3: Extract
- 4: Abstract
- 5: Reorganize
- 6: Release
- 7: Retrospective

0: Explore

Technical feasibility,
hypotheses, border cases

The goal is to learn and verify,
delete when finished

Give yourself a timebox

1: Goal Card

State your goal
Stick it on a card

Don't allow interruptions while working
on a card

1: Goal Card

*Excel sheet with
expressions that refer
to cells in all
directions*

2: Start at the End

Make static "sketch" of the result
before adding dynamic features
(FITYMI)

1 b) dynamic sketches,
e.g all JS, CSS in a single HTML file

2: Start at the End

One "end" can be a test case.

3: Extract

Replace static values with variables

Replace repetitions with mappings and loops.

4: Abstract

Discover the concept behind what you have extracted. Give it a name.

It should work for itself and in combination.

Revert if you cannot find one.

4: Abstract

Consider Types and Interfaces

5: Reorganize

Organize and re-factor to make your future work easier.

Facilitate extensions or improvements.

Prepare for release.

6: Release

The solution must stand on its own without tacit knowledge or external help.

Tests, documentation, examples.

Before every push to the repository.

7: Retrospective

What to keep?

What to try differently next time?

Moves

- 0: Explore
- 1: Goal Card
- 2: Start at the End
- 3: Extract
- 4: Abstract
- 5: Reorganize
- 6: Release
- 7: Retrospective

Observations

Balance and distance

Which technologies support my moves:
dynamic exploration, refactoring

Per feature, per project, whole career, ...