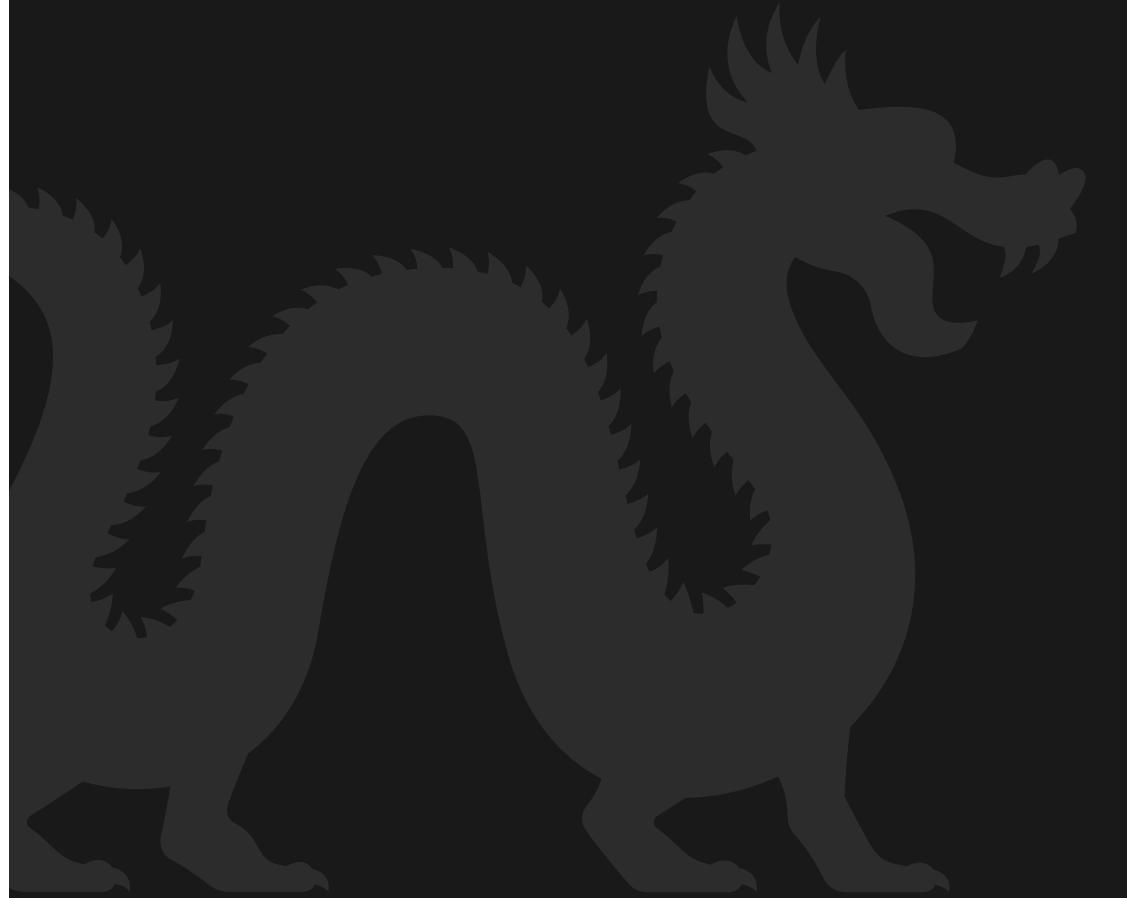


GUIDE TO DX

developer experience

IN BLACK & WHITE



CLEAN | DX

developer experience

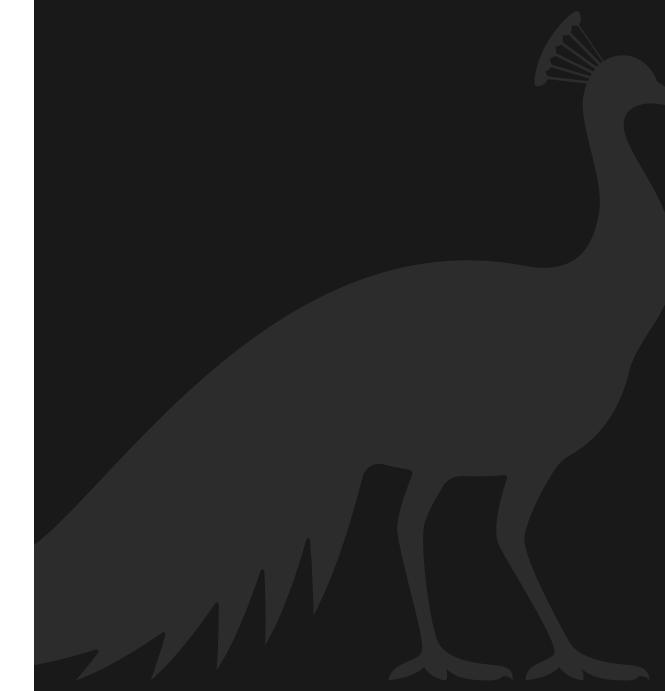
Purpose

- Getting Organized

- Uniformity

- Engineering

Culture



A bird's eye view

CLEAN | DX

developer experience

If you need documentation
then your developer experience
is not intuitive enough



CLEAN | DX

developer experience

The developer experience (**DX**) defines a developer's experience while working within a codebase

The guiding principle of a good DX is a **low cognitive complexity**

In other words: **simplicity**



code cleanliness

ease of debugging

code legibility

maintainability

code organization

intuitive navigation

low cognitive complexity

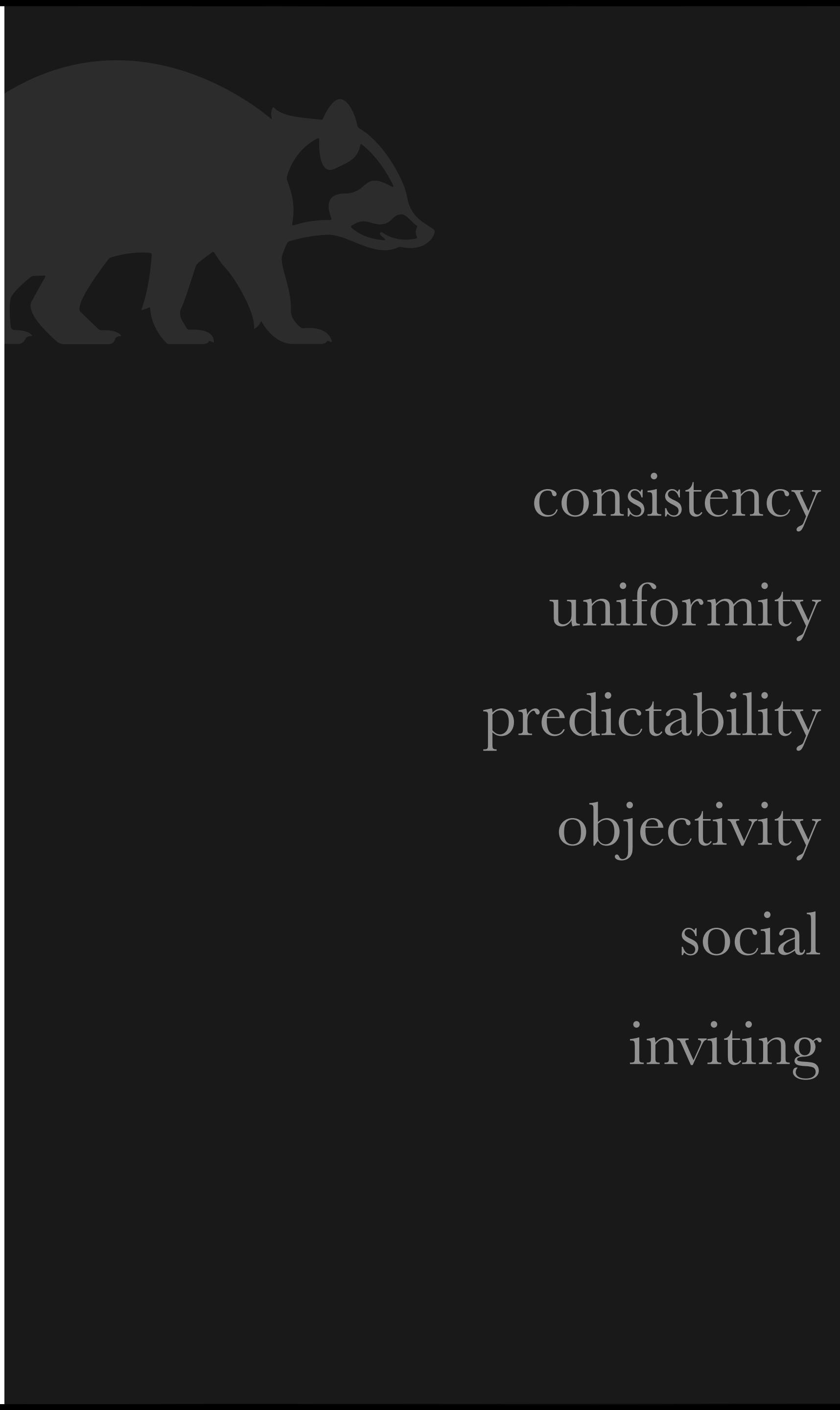
CLEAN | DX

developer experience

Define **clear** conventions that are
easy to follow

Promote common sense and
intuitive solutions

This drives **developer engagement**.



consistency
uniformity
predictability
objectivity
social
inviting

CLEAN | DX

developer experience

Design your solutions to be
easily understood by the most
junior developer



CLEAN | DX

developer experience

Begin with a **code organization** that is both logical and intuitive

Your file system should **make finding things easy** and intuitive even without a search tool.



colocation

discoverability

deletability

simplicity

intuitive layout

CLEAN | DX

developer experience

Keep structures as flat as possible for
as long as possible

Don't make it difficult to find items
by having to traverse multiple levels

Avoid over-thinking groupings.

flat hierarchies
linear
simplicity

CLEAN | DX

developer experience

Having deeply nested hierarchies
makes it more difficult for natural
patterns to emerge

It also makes it more difficult to find
re-usability opportunities.

natural patterns
bubble up
code duplication
scalability
don't repeat yourself

CLEAN | DX

developer experience

Make things so **uniform** that one does not need to know the code to find what one is looking for



CLEAN | DX

developer experience

Reduce variations wherever possible

Don't waste creativity on plumbing such as naming of files. Defer that to a predictable nomenclature

Spend creativity on creative problems.

eliminate waste
predictable
normalize
uniform

spend time wisely

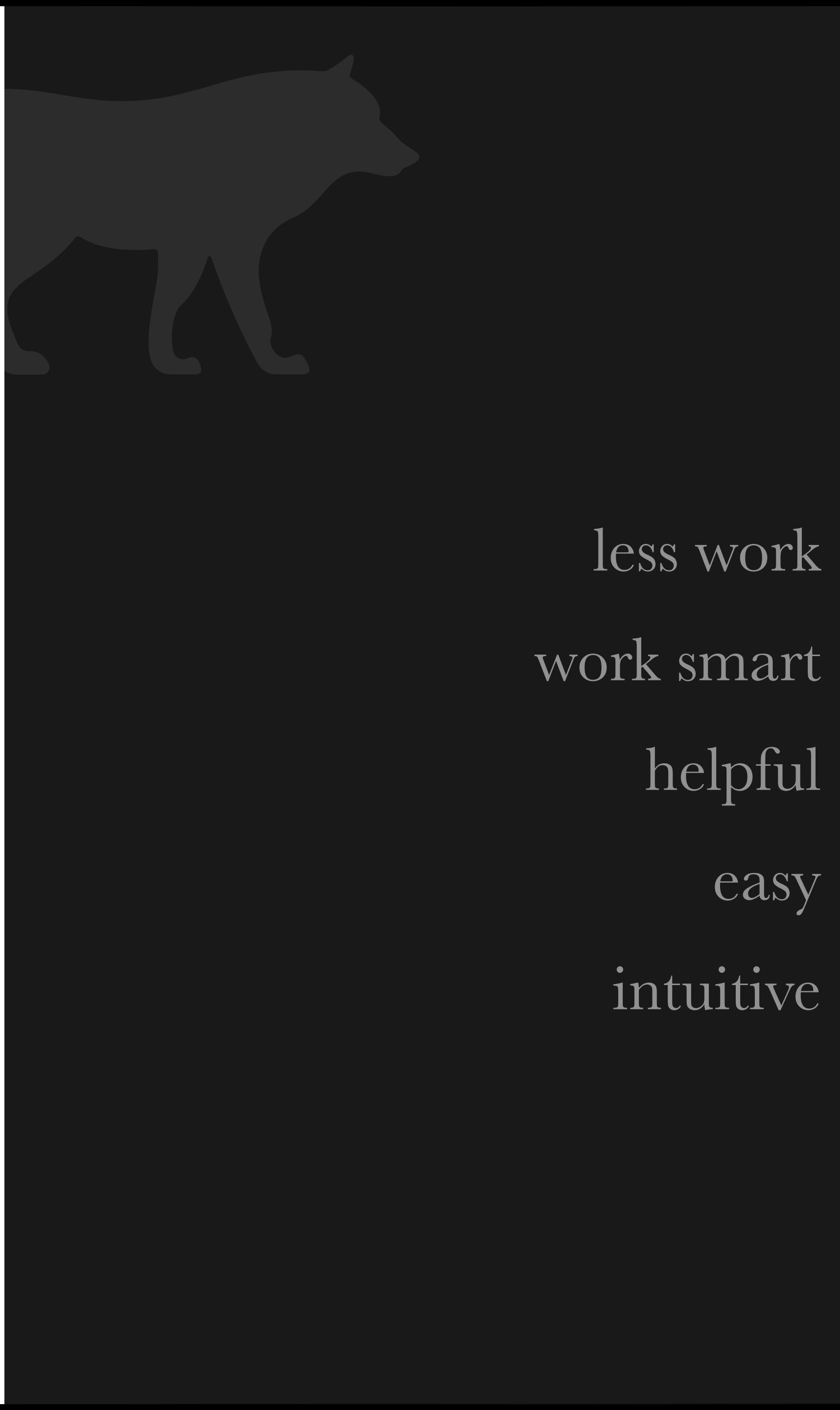
CLEAN | DX

developer experience

Invest effort in automation and tooling

Systems should be self-healing whenever possible

When systems need human assistance, attempt to provide a hand-holding experience where the system guides the user towards a solution.



less work
work smart
helpful
easy
intuitive

CLEAN | DX

developer experience

Encourage collocating technical documentation with codebase

Follow a TL;DR; approach to documentation

Strive for lean documentation and favor intuitive design.



less reading
more doing
action
productivity
no time

CLEAN | DX

developer experience

Predictability has been achieved when a person can guess where to find things in a codebase without the reliance on tribal knowledge, peer assistance or documentation.



self-reliant
self-sufficient
independent
autonomous
anticipation

CLEAN | DX

developer experience

Code with intent



CLEAN | DX

developer experience

Communicate intent with unambiguous naming conventions

Code should read more like a story than code. Non-technical people should be able to derive intent.

code with intent
storyline
layman terms
legible
comprehensible

CLEAN | DX

developer experience

Use **human-friendly** naming conventions. More descriptive names are better than short ambiguous names

Avoid **abbreviations** and acronyms

Leave minification and obfuscation to tooling.



descriptive names
semantics
intent
communication

CLEAN | DX

developer experience

Adopt principles, patterns, and methodologies that reduce trial and error

Continuously observe the emergence of project specific patterns that can be abstracted.



single responsibility
open-closed
liskov substitution
interface
dependency inversion
kiss

CLEAN | DX

developer experience

Do the right thing and you won't have problems



CLEAN | DX

developer experience

Cutting corners and procrastinating proper solutions contribute to higher tech debt

Resist the temptation of making deadlines at the expense of poor foundation building

Communicate technical needs.



communication
expectations
foundation
prepare
engineer for success
avoid tech debt

CLEAN | DX

developer experience

Don't clutter workspace with comments and annotations. Instead focus on making code clear

Remember that the more the need to document, the more likely the code needs to be simplified

Focus on making things discoverable.



sensory overload
overdocument
clutter
cognitive load
distractions

CLEAN | DX

developer experience

Don't commit temporary code because
it will become permanent

Code only for today's needs; tomorrow
is uncertain and will potentially result
in unused code bloat

Add code only when it's needed.



business changes
expired needs
less is more

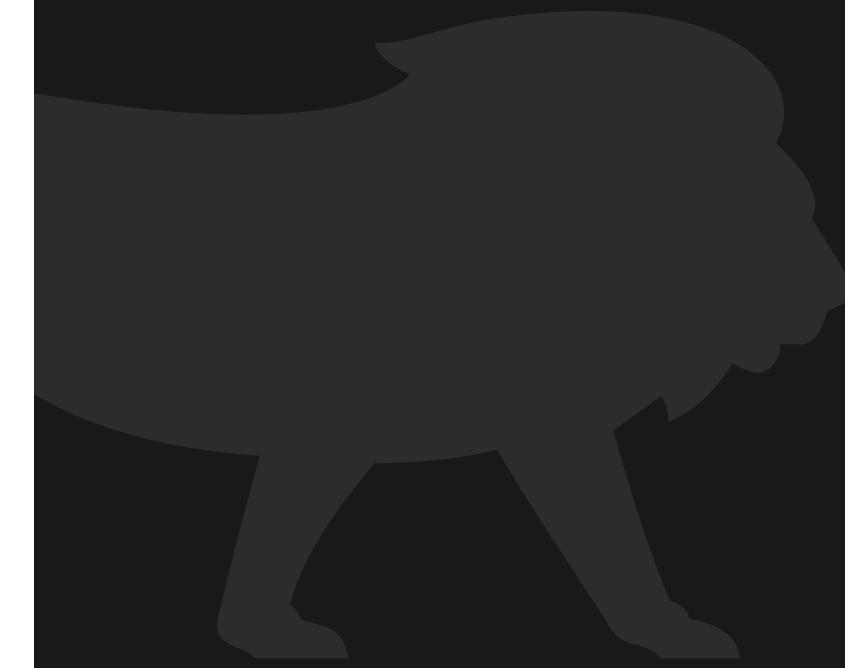
CLEAN | DX

developer experience

Cultivate a culture that emphasizes code formatting and cleanliness even without tooling

This helps with synchronizing the team's attention to the same detail and way of thinking

Respect formatting the same as code.



interactions
homogeny
synchronization
all for one
one for all
mindset

CLEAN | DX

developer experience

Prefer conventions over tooling as it helps develop a collective team culture that in turn helps synchronize the approach of reasoning

This helps shape a thought process that is less messy and therefore more predictable.

discipline
coordination
collaboration
cohesion

CLEAN | DX

developer experience

Eliminate developer's fears by
making things clear



CLEAN | DX

developer experience

Consistency

Cleanliness

Clarity

Quality

Simplicity

Take-aways

CLEAN | DX

developer experience

Milan has over 27 years of professional experience

Presently he focuses on front-end technologies, especially React

Contact him for engagements.



CLEAN | DX

developer experience

Website coming soon...

www.cleandx.com

