Naming convention

Methods prefixed with:

- `as_` - takes self by reference and return a reference; should be cheap; e.g.: `as_str()`, `as_slice()`

- `to_` - takes self by reference and returns new value; may be expensive; e.g.: `to_owned()`, `to_bytes()`

- `into_` - takes self by value (move/consume) and return new value; it may or may not be cheap; e.g: `into()`, `into_iter()`, `into_vec()`

- `try_` - methods that may fail/return `Result`; e.g.: `try_unwrap()`, `try_into()`

! type (RFC 1216)

`!` represents empty type that has no value:

- they never exist at runtime because there is no way to create one
- they have no logical machine-level representation
- code that handles them never executes
- represent the return type of functions that never return
- can be converted to any other type
- e.g.: `baz() -> !`, `foo() -> Result<!, io::Error>`, `bar() -> Result<u32, !>`