Short programming scheme

```
          ┌─────────────────┐
     ┌───▶│      Main       │
     │    └────────┬────────┘
     │             │
     │             ▼
     │    ┌─────────────────┐          ┌──────────────────────────┐
     │    │ init_minishell()│ ─ ─ ─ ▶  │      t_minishell         │
     │    └─────────────────┘          ├──────────────────────────┤
     │                                 │   char *hostname         │
     │    ┌──────────────────────────┐ └──────────────────────────┘
     │    │ While true               │
     │    ├──────────────────────────┤
     │    │  ┌─────────────────┐     │
     └──────▶│ prompt(t_minishell│ ─ ─ ─ ▶ @param t_minishell minishell
            │  │       *)        │          - readline with custom
            │  └────────┬────────┘          prompt
            │           │                   - add line to history when
            │           ▼                   valid
            │      ◇ is prompt NULL ? ◇      @returns char*
            │                    │ yes
            │      ┌─────────┐   │
            │      │  break  │◀──┘
            │      └─────────┘
            └──────────────────────────┘
```

null means error
prompt while repeat itself if
readline is null, it will never
return a NULL pointer or
white spaces only
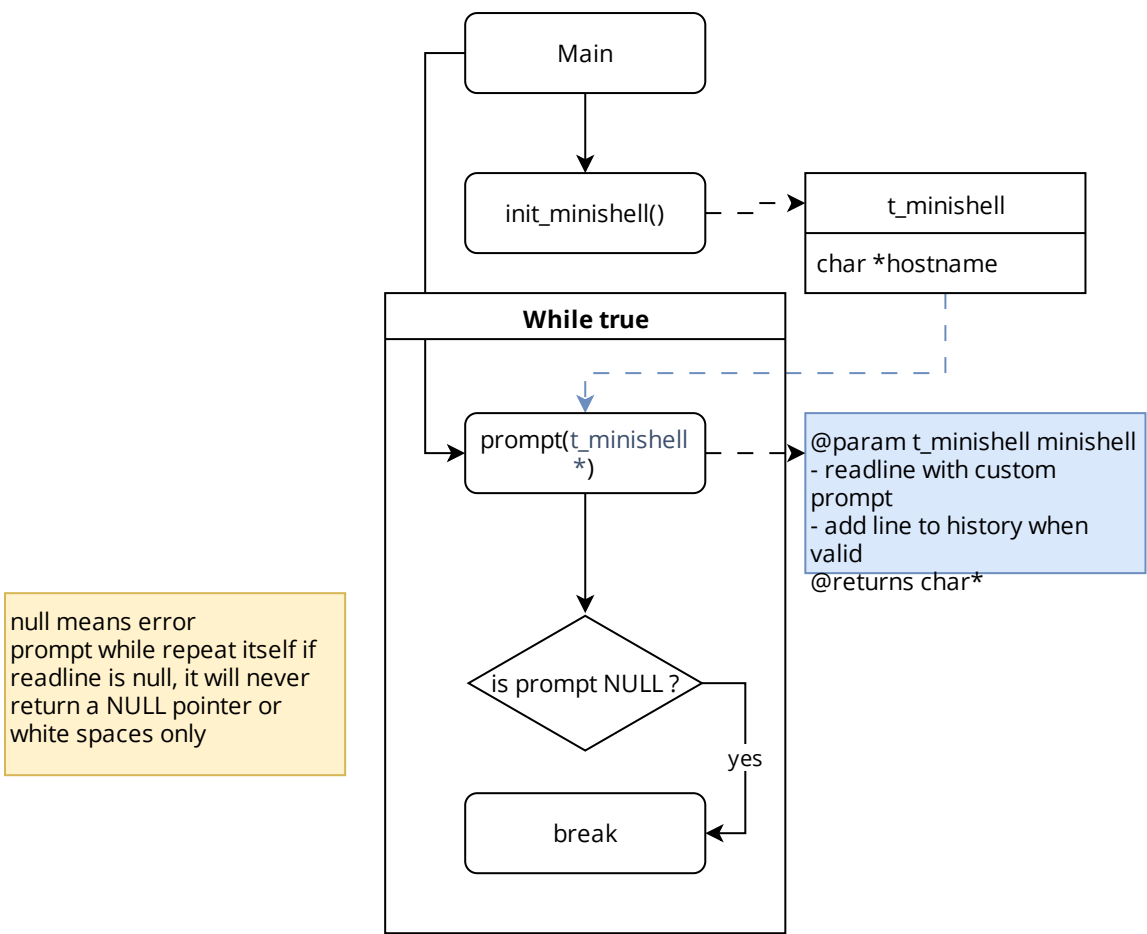
Data structures

t_minishell is the main data structure

```
┌──────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│   t_minishell    │      │   struct t_env   │      │  enum t_env_type │
├──────────────────┤      ├──────────────────┤      ├──────────────────┤
│ char *hostname   │      │ char *name       │      │ type_string      │
│                  │ ───▶ │                  │ ───▶ │                  │
│ t_env *envs      │      │ void *value      │      │ type_num         │
└──────────────────┘      │                  │      │                  │
                          │ t_env_type type  │      │ type_token       │
                          │                  │      └──────────────────┘
                          │ t_env *prev      │
                          │                  │
                          │ t_env *next      │
                          └──────────────────┘
```