FLOW SCHEME

Short programming scheme

```
                    ┌──────────────┐
                    │     Main     │
                    └──────────────┘
                            │
                    ┌──────────────┐
                    │init_minishell()│- - - - -
                    └──────────────┘
          ┌─────────────────────────────────────────┐
          │ While true                               │
          │                                          │
          │         ┌──────────────┐                 │
          │    ┌───▶│prompt(t_minishell *)│- - - - -▶
          │    │    └──────────────┘
          │    │           │
          │    │     ◇ is prompt NULL ? ◇──── yes
          │    │           │
          │    │           no           break
          │    │    ┌──────────────┐
          │    │    │lexer(t_minishell *, input)│
          │    │    └──────────────┘
          │    │    ┌──────────────┐
          │    │    │parsing(t_minishell *, lex)│
          │    │    └──────────────┘
          │    │    ┌──────────────┐
          │    └────│execute(t_token *tokens)│
          │         └──────────────┘
          └─────────────────────────────────────────┘
```
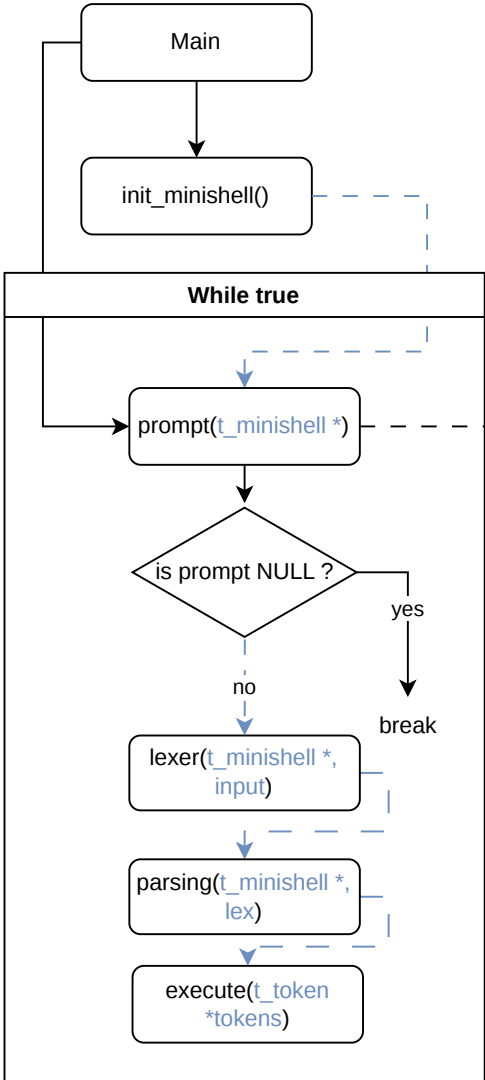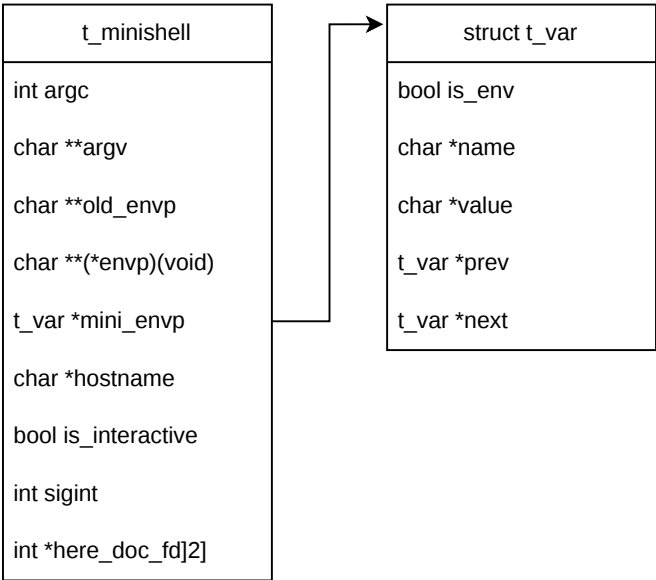
the short programming scheme may not be updated and relevant anymore

@param t_minishell minishell
- readline with custom prompt
- add line to history when valid
@returns char*

Data structures

minishell & variables

t_minishell
- int argc
- char **argv
- char **old_envp
- char **(*envp)(void)
- t_var *mini_envp
- char *hostname
- bool is_interactive
- int sigint
- int *here_doc_fd]2]

struct t_var
- bool is_env
- char *name
- char *value
- t_var *prev
- t_var *next

tokenization

struct t_token
- void *data
- t_token_type type
- t_token *next

enum t_token_type
- token_cmd
- token_var
- token_pipe
- token_stdin
- token_stdout
- token_and
- token_or
- token_subshell

struct t_word
- char *str
- bool is_var
- bool is_quoted
- t_word *next

t_type == structure type to deference with

typed tokens structures

struct t_cmd
- t_word *cmd
- t_word **argv
- int argc
- int status
- pid_t pid

struct t_pipe
- int[2] pipe

token_and
- no structure needed

token_or
- no structure needed

struct t_stdout
- bool is_append
- t_word *filename

struct t_var_init
- char *name
- t_word *value

struct t_stdin
- bool is_heredoc
- bool is_quoted
- char *limiter
- t_word *filename

t_subshell
- t_token *token
- pid_t pid