Stdlib	Containers	Batteries	Base
	CCOpt.(<$>) : ('a -> 'b) -> 'a option -> 'b option		
	CCOpt.(<*>) : ('a -> 'b) option -> 'a option -> 'b option		
	CCOpt.(<+>) : 'a option -> 'a option -> 'a option		
	CCOpt.(>>=) : 'a option -> ('a -> 'b option) -> 'b option		Base.Option.(>>=) : 'a option -> ('a -> 'b option) -> 'b option
	CCOpt.(>|=) : 'a option -> ('a -> 'b) -> 'b option		Base.Option.(>>|) : 'a option -> ('a -> 'b) -> 'b option
		BatOption.(|?) : 'a option -> 'a -> 'a	
	CCOpt.(and*) : 'a option -> 'b option -> ('a * 'b) option		
	CCOpt.(and+) : 'a option -> 'b option -> ('a * 'b) option		
	CCOpt.(let*) : 'a option -> ('a -> 'b option) -> 'b option		
	CCOpt.(let+) : 'a option -> ('a -> 'b) -> 'b option		
		BatOption.Labels.map : f:('a -> 'b) -> 'a option -> 'b option	
		BatOption.Labels.map_default : f:('a -> 'b) -> 'b -> 'a option -> 'b	
		BatOption.Labels.may : f:('a -> unit) -> 'a option -> unit	
Option.None : 'a option = Option.None			Base.Option.None : 'a option = Base.Option.None
			Base.Option.all : 'a option list -> 'a list option
			Base.Option.all_unit : unit option list -> unit option
		BatOption.apply : ('a -> 'a) option -> 'a -> 'a	
Option.bind : 'a option -> ('a -> 'b option) -> 'b option	CCOpt.bind : 'a option -> ('a -> 'b option) -> 'b option	BatOption.bind : 'a option -> ('a -> 'b option) -> 'b option	Base.Option.bind : 'a option -> f:('a -> 'b option) -> 'b option
			Base.Option.both : 'a option -> 'b option -> ('a * 'b) option
			Base.Option.call : 'a -> f:('a -> unit) option -> unit
	CCOpt.choice : 'a option list -> 'a option		
	CCOpt.choice_iter : 'a option CCOpt.iter -> 'a option		
	CCOpt.choice_seq : 'a option Seq.t -> 'a option		
Option.compare : ('a -> 'a -> int) -> 'a option -> 'a option -> int	CCOpt.compare : ('a -> 'a -> int) -> 'a option -> 'a option -> int	BatOption.compare : ?cmp:('a -> 'a -> int) -> 'a option -> 'a option -> int	Base.Option.compare : ('a -> 'a -> int) -> 'a option -> 'a option -> int
			Base.Option.count : 'a option -> f:('a -> bool) -> int
		BatOption.default : 'a -> 'a option -> 'a	
		BatOption.default_delayed : (unit -> 'a) -> 'a option -> 'a	
		BatOption.enum : 'a option -> 'a BatEnum.t	
		BatOption.eq : ?eq:('a -> 'a -> bool) -> 'a option -> 'a option -> bool	
Option.equal : ('a -> 'a -> bool) -> 'a option -> 'a option -> bool	CCOpt.equal : ('a -> 'a -> bool) -> 'a option -> 'a option -> bool		Base.Option.equal : 'a Base.Equal.equal -> 'a option Base.Equal.equal
	CCOpt.exists : ('a -> bool) -> 'a option -> bool		Base.Option.exists : 'a option -> f:('a -> bool) -> bool
	CCOpt.filter : ('a -> bool) -> 'a option -> 'a option	BatOption.filter : ('a -> bool) -> 'a option -> 'a option	Base.Option.filter : 'a option -> f:('a -> bool) -> 'a option
			Base.Option.find : 'a option -> f:('a -> bool) -> 'a option
	CCOpt.flat_map : ('a -> 'b option) -> 'a option -> 'b option		Base.Option.find_map : 'a option -> f:('a -> 'b option) -> 'b option

Stdlib	Containers	Batteries	Base
			Base.Option.first_some : 'a option -> 'a option -> 'a option
Option.fold : none:'a -> some:('b -> 'a) -> 'b option -> 'a	CCOpt.fold : ('a -> 'b -> 'a) -> 'a -> 'b option -> 'a		Base.Option.fold : 'a option -> init:'accum -> f:('accum -> 'a -> 'accum) -> 'accum
			Base.Option.fold_result : 'a option -> init:'accum -> f:('accum -> 'a -> ('accum, 'e) Base.Result.t) -> ('accum, 'e) Base.Result.t
			Base.Option.fold_until : 'a option -> init:'accum -> f:('accum -> 'a -> ('accum, 'final) Base.Container_intf.Continue_or_stop.t) -> finish:('accum -> 'final) -> 'final
	CCOpt.for_all : ('a -> bool) -> 'a option -> bool		Base.Option.for_all : 'a option -> f:('a -> bool) -> bool
Option.get : 'a option -> 'a		BatOption.get : 'a option -> 'a	
		BatOption.get_exn : 'a option -> exn -> 'a	
	CCOpt.get_exn_or : string -> 'a option -> 'a		
	CCOpt.get_lazy : (unit -> 'a) -> 'a option -> 'a		
	CCOpt.get_or : default:'a -> 'a option -> 'a		
			Base.Option.hash_fold_t : (Base.Ppx_hash_lib.Std.Hash.state -> 'a -> Base.Ppx_hash_lib.Std.Hash.state) -> Base.Ppx_hash_lib.Std.Hash.state -> 'a option -> Base.Ppx_hash_lib.Std.Hash.state
	CCOpt.if_ : ('a -> bool) -> 'a -> 'a option		
			Base.Option.ignore_m : 'a option -> unit option
			Base.Option.invariant : 'a Base.Invariant_intf.inv -> 'a option Base.Invariant_intf.inv
			Base.Option.is_empty : 'a option -> bool
Option.is_none : 'a option -> bool	CCOpt.is_none : 'a option -> bool	BatOption.is_none : 'a option -> bool	Base.Option.is_none : 'a option -> bool
Option.is_some : 'a option -> bool	CCOpt.is_some : 'a option -> bool	BatOption.is_some : 'a option -> bool	Base.Option.is_some : 'a option -> bool
Option.iter : ('a -> unit) -> 'a option -> unit	CCOpt.iter : ('a -> unit) -> 'a option -> unit		Base.Option.iter : 'a option -> f:('a -> unit) -> unit
Option.join : 'a option option -> 'a option	CCOpt.flatten : 'a option option -> 'a option		Base.Option.join : 'a option option -> 'a option
			Base.Option.length : 'a option -> int
Option.map : ('a -> 'b) -> 'a option -> 'b option	CCOpt.map : ('a -> 'b) -> 'a option -> 'b option	BatOption.map : ('a -> 'b) -> 'a option -> 'b option	Base.Option.map : 'a option -> f:('a -> 'b) -> 'b option
	CCOpt.map2 : ('a -> 'b -> 'c) -> 'a option -> 'b option -> 'c option		Base.Option.map2 : 'a option -> 'b option -> f:('a -> 'b -> 'c) -> 'c option
		BatOption.map_default : ('a -> 'b) -> 'b -> 'a option -> 'b	
		BatOption.map_default_delayed : ('a -> 'b) -> (unit -> 'b) -> 'a option -> 'b	
	CCOpt.map_lazy : (unit -> 'b) -> ('a -> 'b) -> 'a option -> 'b		
	CCOpt.map_or : default:'b -> ('a -> 'b) -> 'a option -> 'b		
			Base.Option.max_elt : 'a option -> compare:('a -> 'a -> int) -> 'a option
		BatOption.may : ('a -> unit) -> 'a option -> unit	
			Base.Option.mem : 'a option -> 'a -> equal:('a -> 'a -> bool) -> bool
			Base.Option.merge : 'a option -> 'a option -> f:('a -> 'a -> 'a) -> 'a option
			Base.Option.min_elt : 'a option -> compare:('a -> 'a -> int) -> 'a option
		BatOption.of_enum : 'a BatEnum.t -> 'a option	
	CCOpt.of_list : 'a list -> 'a option		
	CCOpt.of_result : ('a, 'b) result -> 'a option		
	CCOpt.or_ : else_:'a option -> 'a option -> 'a option		

Stdlib	Containers	Batteries	Base
	CCOpt.or_lazy : else_:(unit -> 'a option) -> 'a option -> 'a option		
		BatOption.ord : 'a BatOrd.ord -> 'a option BatOrd.ord	
	CCOpt.pp : 'a CCOpt.printer -> 'a option CCOpt.printer		
		BatOption.print : ('a BatInnerIO.output -> 'b -> unit) -> 'a BatInnerIO.output -> 'b option -> unit	
	CCOpt.pure : 'a -> 'a option		
	CCOpt.random : 'a CCOpt.random_gen -> 'a option CCOpt.random_gen		
	CCOpt.return : 'a -> 'a option		Base.Option.return : 'a -> 'a option
	CCOpt.return_if : bool -> 'a -> 'a option		
	CCOpt.sequence_l : 'a option list -> 'a list option		
			Base.Option.sexp_of_t : ('a -> Sexplib0__.Sexp.t) -> 'a option -> Sexplib0__.Sexp.t
		BatOption.some : 'a -> 'a option	Base.Option.some : 'a -> 'a option
			Base.Option.some_if : bool -> 'a -> 'a option
			Base.Option.sum : (module Base.Container_intf.Summable with type t = 'sum) -> 'a option -> f:('a -> 'sum) -> 'sum
			Base.Option.t_of_sexp : (Sexplib0__.Sexp.t -> 'a) -> Sexplib0__.Sexp.t -> 'a option
			Base.Option.t_sexp_grammar : Base.Ppx_sexp_conv_lib...
			Base.Option.to_array : 'a option -> 'a array
	CCOpt.to_gen : 'a option -> 'a CCOpt.gen		
	CCOpt.to_iter : 'a option -> 'a CCOpt.iter		
Option.to_list : 'a option -> 'a list	CCOpt.to_list : 'a option -> 'a list		Base.Option.to_list : 'a option -> 'a list
Option.to_result : none:'e -> 'a option -> ('a, 'e) result	CCOpt.to_result : 'e -> 'a option -> ('a, 'e) result		
	CCOpt.to_result_lazy : (unit -> 'e) -> 'a option -> ('a, 'e) result		
Option.to_seq : 'a option -> 'a Seq.t	CCOpt.to_seq : 'a option -> 'a Seq.t		
			Base.Option.try_with : (unit -> 'a) -> 'a option
			Base.Option.try_with_join : (unit -> 'a option) -> 'a option
			Base.Option.validate : none:unit Base.Validate.check -> some:'a Base.Validate.check -> 'a option Base.Validate.check
Option.value : 'a option -> default:'a -> 'a	CCOpt.value : 'a option -> default:'a -> 'a		Base.Option.value : 'a option -> default:'a -> 'a
			Base.Option.value_exn : ?here:Base.Source_code_position0.t -> ?error:Base.Error.t -> ?message:string -> 'a option -> 'a
			Base.Option.value_map : 'a option -> default:'b -> f:('a -> 'b) -> 'b
	CCOpt.wrap : ?handler:(exn -> bool) -> ('a -> 'b) -> 'a -> 'b option		
	CCOpt.wrap2 : ?handler:(exn -> bool) -> ('a -> 'b -> 'c) -> 'a -> 'b -> 'c option		
			Base.Option_array.blit : ('a Base.Option_array.t, 'a Base.Option_array.t) Base.Blit_intf.blit
			Base.Option_array.blito : ('a Base.Option_array.t, 'a Base.Option_array.t) Base.Blit_intf.blito
			Base.Option_array.clear : 'a Base.Option_array.t -> unit
			Base.Option_array.copy : 'a Base.Option_array.t -> 'a Base.Option_array.t
			Base.Option_array.create : len:int -> 'a Base.Option_array.t
			Base.Option_array.empty : 'a Base.Option_array.t

Stdlib	Containers	Batteries	Base
			Base.Option_array.get : 'a Base.Option_array.t -> int -> 'a option
			Base.Option_array.get_some_exn : 'a Base.Option_array.t -> int -> 'a
			Base.Option_array.init : int -> f:(int -> 'a option) -> 'a Base.Option_array.t
			Base.Option_array.init_some : int -> f:(int -> 'a) -> 'a Base.Option_array.t
			Base.Option_array.is_none : 'a Base.Option_array.t -> int -> bool
			Base.Option_array.is_some : 'a Base.Option_array.t -> int -> bool
			Base.Option_array.length : 'a Base.Option_array.t -> int
			Base.Option_array.set : 'a Base.Option_array.t -> int -> 'a option -> unit
			Base.Option_array.set_none : 'a Base.Option_array.t -> int -> unit
			Base.Option_array.set_some : 'a Base.Option_array.t -> int -> 'a -> unit
			Base.Option_array.sexp_of_t : ('a -> Sexplib0__.Sexp.t) -> 'a Base.Option_array.t -> Sexplib0__.Sexp.t
			Base.Option_array.sub : ('a Base.Option_array.t, 'a Base.Option_array.t) Base.Blit_intf.sub
			Base.Option_array.subo : ('a Base.Option_array.t, 'a Base.Option_array.t) Base.Blit_intf.subo
			Base.Option_array.swap : 'a Base.Option_array.t -> int -> int -> unit
			Base.Option_array.t_of_sexp : (Sexplib0__.Sexp.t -> 'a) -> Sexplib0__.Sexp.t -> 'a Base.Option_array.t
			Base.Option_array.unsafe_blit : ('a Base.Option_array.t, 'a Base.Option_array.t) Base.Blit_intf.blit
			Base.Option_array.unsafe_get : 'a Base.Option_array.t -> int -> 'a option
			Base.Option_array.unsafe_get_some_assuming_some : 'a Base.Option_array.t -> int -> 'a
			Base.Option_array.unsafe_get_some_exn : 'a Base.Option_array.t -> int -> 'a
			Base.Option_array.unsafe_is_some : 'a Base.Option_array.t -> int -> bool
			Base.Option_array.unsafe_set : 'a Base.Option_array.t -> int -> 'a option -> unit
			Base.Option_array.unsafe_set_none : 'a Base.Option_array.t -> int -> unit
			Base.Option_array.unsafe_set_some : 'a Base.Option_array.t -> int -> 'a -> unit