

# DLParse

kb : Optional[KnowledgeBase]  
queries\_list : list[Query]

\_check\_abstract(c: Concept): None  
\_create\_fuzzy\_number(tokens: pp.ParseResults): pp.ParseResults  
\_fuzzy\_logic\_parser(tokens: pp.ParseResults): pp.ParseResults  
\_get\_modifier(m: str): Modifier  
\_parse\_axioms(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_binary\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_constraints(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_crisp\_declarations(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_datatype\_restriction(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_degree(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_expression(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_feature(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_fuzzy\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_fuzzy\_equivalence(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_fuzzy\_number\_range(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_fuzzy\_similarity(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_inequation(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_modifier(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_modifier\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_owa\_integral\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_q\_owa\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_queries(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_restrictions(): typing.Any  
\_parse\_sigma\_count\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_term(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_threshold\_concept(tokens: pp.ParseResults)  
\_parse\_truth\_constants(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_unary\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_weighted\_concept(tokens: pp.ParseResults): pp.ParseResults  
\_parse\_weighted\_concept\_simple(tokens: pp.ParseResults): pp.ParseResults  
\_set\_fuzzy\_number(tokens: pp.ParseResults): pp.ParseResults  
\_show\_abstract\_fillers(tokens: pp.ParseResults): pp.ParseResults  
\_show\_abstract\_fillers\_for(tokens: pp.ParseResults): pp.ParseResults  
\_show\_concepts(tokens: pp.ParseResults): pp.ParseResults  
\_show\_concrete\_fillers(tokens: pp.ParseResults): pp.ParseResults  
\_show\_concrete\_fillers\_for(tokens: pp.ParseResults): pp.ParseResults  
\_show\_concrete\_instance\_for(tokens: pp.ParseResults): pp.ParseResults  
\_show\_instances(tokens: pp.ParseResults): pp.ParseResults  
\_show\_languages(tokens: pp.ParseResults): pp.ParseResults  
\_show\_variables(tokens: pp.ParseResults): pp.ParseResults  
\_to\_concept(c: typing.Union[str, Concept]): Concept  
\_to\_number(tokens: pp.ParseResults): float | int  
\_to\_top\_bottom\_concept(tokens: pp.ParseResults): pp.ParseResults  
get\_grammatics(): pp.ParserElement  
get\_kb(): tuple[KnowledgeBase, list[Query]]  
load\_config(): None  
main(): None  
parse\_string(instring: str): pp.ParseResults  
parse\_string\_opt(filename: str): pp.ParseResults