

GraphQL	Java	JavaScript (or TypeScript)
input (<i>SDL keyword</i>)	@InputOnly public final class	prototypical type, prototype
type (<i>SDL keyword</i>)	@OutputOnly public final class	prototypical type, prototype
interface (<i>SDL keyword</i>)	@OutputOnly interface	~one of prototype's prototypes
union (<i>SDL keyword</i>)	Approximately an empty marker interface implemented by relevant classes.	n/a, unions in TypeScript
"field" In a query In a mutation In a subscription	method (call) Non-void, no side-effects Non-void with side-effects Void notification control	prototypical function (call) returning something, no side-effects returning something, w/ side-effects no return, notification control
query (root field)	Predefined, named, statically imported singleton of the Query type.	Predefined, named, singleton of the Query type.
Query (built-in type)	The class of the query singleton.	The prototype of the query singleton.
mutation (root field)	Predefined, named, statically imported singleton of the Mutation type.	Predefined, named, singleton of the Mutation type.
Mutation (built-in type)	The class of the mutation singleton.	The prototype of the mutation singleton.
subscription (root field)	Predefined, named, statically imported singleton of the Subscription type.	Predefined, named, singleton of the Subscription type.
Subscription (built-in type)	The class of the subscription singleton.	The prototype of the subscription singleton.
"scalar" (<i>built-in</i>) Boolean Int Float String ID	boxed (nullable) primitive type Boolean Integer Double String ~ Integer or UUID or String	primitive type Boolean ~ Number (<i>32-bit int subset</i>) Number String ~ Number or String
scalar (<i>keyword</i>)	Custom primitive-like value type that can be represented using one of the primitive type literals in JSON.	
enum	enum	n/a, enum in TypeScript
directive	annotation, without runtime retention (not reflectable)	n/a, decorators in TypeScript
argument	method or annotation parameter and/or argument	Function or decorator (TypeScript) parameter and/or argument
variable	request method argument	request function argument
!	@NonNull / @NotNull	n/a, NonNullable in TypeScript
{	.map(_ -> new JsonObject<>() {	.map((_) => {
introspection	reflection	reflection
error	exception, throwable or error	exception