

module-stmt

```

module module-name {
  yang-version 1;
  namespace "uri-string";
  prefix "prefix-string";

  // zero or more imports
  import "module-name" {
    prefix "prefix-string";
    revision-date "date-string";
  }

  // zero or more includes
  include "submodule-name" {
    revision-date "date-string";
  }

  organization "org-name";
  contact "contact info";
  description "module description";
  reference "module reference";

  // zero or more revision identifiers
  revision "date-string" {
    description "revision description";
    reference "revision reference";
  }

  body-stmt; // zero or more
}

```

data-def-stmt

```

container name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  must "xpath-expr"; // zero or more
  presence "presence description";
  config true/false;
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  typedef-stmt; // zero or more
  grouping-stmt; // zero or more
  data-def-stmt; // zero or more
}

leaf name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  type-stmt;
  units "units name";
  must "xpath-expr"; // zero or more
  default "default value";
  config true/false;
  mandatory true/false; // no default if true
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

```

body-stmt

```

extension extension-name {
  argument argument-name {
    yin-element true/false;
  }
}

feature feature-name {
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

identity identity-name {
  base identity-base; // no base == root
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

typedef type-name {
  type-stmt;
  units "units name";
  default "default value";
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

grouping grouping-name {
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  typedef-stmt; // zero or more
  grouping-stmt; // zero or more
  data-def-stmt; // zero or more
}

[data-def-stmt]

augment "/path/to/augment/target" {
  when "when-xpath-expr";
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  data-def-stmt | case-def-stmt; // one or more
}

rpc rpc-name {
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  input {
    typedef-stmt; // zero or more
    grouping-stmt; // zero or more
    data-def-stmt; // one or more
  }
  output {
    typedef-stmt; // zero or more
    grouping-stmt; // zero or more
    data-def-stmt; // one or more
  }
}

notification notification-name {
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  typedef-stmt; // zero or more
  grouping-stmt; // zero or more
  data-def-stmt; // zero or more
}

```

```

leaf-list name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  type-stmt;
  units "units name";
  must "xpath-expr"; // zero or more
  config true/false;
  min-elements number; // default zero
  max-elements number | unbounded;
  ordered-by user | system;
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

```

```

list name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  must "xpath-expr"; // zero or more
  key "key leafs"; // req. if config=true
  unique "unique-leafs"; // zero or more
  config true/false;
  min-elements number; // default zero
  max-elements number | unbounded;
  ordered-by user | system;
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  typedef-stmt; // zero or more
  grouping-stmt; // zero or more
  data-def-stmt; // one or more
}

```

```

choice name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  default "case name";
  config true/false;
  mandatory true/false;
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  case-stmt | short-case-stmt; // zero or more
}

```

```

case name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  data-def-stmt; // zero or more
}

```

```

anyxml name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  must "xpath-expr"; // zero or more
  config true/false;
  mandatory true/false;
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

```

```

uses grouping-name {
  when "xpath-expr";
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  refine-stmt; / zero or more
  uses-augment-stmt; / zero or more
}

```