Annex A

Grammar Summary

(informative)

A.1 Lexical Grammar

[*SourceCharacter*](#158) ::

any Unicode code point

[*InputElementDiv*](#160) ::

[*WhiteSpace*](#162)

[*LineTerminator*](#162)

[*Comment*](#163)

[*CommonToken*](#163)

[*DivPunctuator*](#166)

[*RightBracePunctuator*](#166)

[*InputElementRegExp*](#160) ::

[*WhiteSpace*](#162)

[*LineTerminator*](#162)

[*Comment*](#163)

[*CommonToken*](#163)

[*RightBracePunctuator*](#166)

[*RegularExpressionLiteral*](#172)

[*InputElementRegExpOrTemplateTail*](#160) ::

[*WhiteSpace*](#162)

[*LineTerminator*](#162)

[*Comment*](#163)

[*CommonToken*](#163)

[*RegularExpressionLiteral*](#172)

[*TemplateSubstitutionTail*](#174)

[*InputElementTemplateTail*](#160) ::

[*WhiteSpace*](#162)

[*LineTerminator*](#162)

[*Comment*](#163)

[*CommonToken*](#163)

[*DivPunctuator*](#166)

[*TemplateSubstitutionTail*](#174)

[*WhiteSpace*](#162) ::

<TAB>

<VT>

<FF>

<SP>

<NBSP>

<ZWNBSP>

<USP>

[*LineTerminator*](#162) ::

<LF>

<CR>

<LS>

<PS>

[*LineTerminatorSequence*](#162) ::

<LF>

<CR>[lookahead ≠ <LF>]

<LS>

<PS>

<CR><LF>

[*Comment*](#163) ::

[*MultiLineComment*](#163)

[*SingleLineComment*](#163)

[*MultiLineComment*](#163) ::

/\**MultiLineCommentChars*opt \*/

[*MultiLineCommentChars*](#163) ::

[*MultiLineNotAsteriskChar*](#163)*MultiLineCommentChars*opt

\**PostAsteriskCommentChars*opt

[*PostAsteriskCommentChars*](#163) ::

[*MultiLineNotForwardSlashOrAsteriskChar*](#163)*MultiLineCommentChars*opt

\**PostAsteriskCommentChars*opt

[*MultiLineNotAsteriskChar*](#163) ::

[*SourceCharacter*](#158) but not \*

[*MultiLineNotForwardSlashOrAsteriskChar*](#163) ::

[*SourceCharacter*](#158) but not one of /  or \*

[*SingleLineComment*](#163) ::

//*SingleLineCommentChars*opt

[*SingleLineCommentChars*](#163) ::

[*SingleLineCommentChar*](#163)*SingleLineCommentChars*opt

[*SingleLineCommentChar*](#163) ::

[*SourceCharacter*](#158) but not [*LineTerminator*](#162)

[*CommonToken*](#163) ::

[*IdentiᲪierName*](#164)

[*Punctuator*](#166)

[*NumericLiteral*](#167)

[*StringLiteral*](#169)

[*Template*](#173)

[*IdentiᲪierName*](#164) ::

[*IdentiᲪierStart*](#164)

[*IdentiᲪierName*](#164)[*IdentiᲪierPart*](#164)

[*IdentiᲪierStart*](#164) ::

[*UnicodeIDStart*](#164)

$

\_

\[*UnicodeEscapeSequence*](#170)

[*IdentiᲪierPart*](#164) ::

[*UnicodeIDContinue*](#165)

$

\_

\[*UnicodeEscapeSequence*](#170)

<ZWNJ>

<ZWJ>

[*UnicodeIDStart*](#164) ::

any Unicode code point with the Unicode property “ID\_Start”

[*UnicodeIDContinue*](#165) ::

any Unicode code point with the Unicode property “ID\_Continue”

[*ReservedWord*](#165) ::

[*Keyword*](#165)

[*FutureReservedWord*](#166)

[*NullLiteral*](#166)

[*BooleanLiteral*](#166)

[*Keyword*](#165) :: one of

break do in typeof case else instanceof var catch export new void class extends return while

const finally super with continue for switch yield debugger function this default if

throw delete import try

[*FutureReservedWord*](#166) ::

enum

await

await is only treated as a *FutureReservedWord* when *Module* is the goal symbol of the syntactic grammar.

The following tokens are also considered to be *FutureReservedWord*s when parsing [strict mode code:](#159)

implements  package  protected

interface  private  public

[*Punctuator*](#166) :: one of

{ ( ) [ ] . ...  ; , < > <= >= == != === !== + ‐ \* % ++ ‐‐ << >> >>> & | ^ ! ~ && || ? : = +=

‐= \*= %= <<= >>= >>>= &= |= ^= => \*\* \*\*=

[*DivPunctuator*](#166) ::

/

/=

[*RightBracePunctuator*](#166) ::

}

[*NullLiteral*](#166) ::

null

[*BooleanLiteral*](#166) ::

true

false

[*NumericLiteral*](#167) ::

[*DecimalLiteral*](#167)

[*BinaryIntegerLiteral*](#167)

[*OctalIntegerLiteral*](#167)

[*HexIntegerLiteral*](#168)

[*DecimalLiteral*](#167) ::

[*DecimalIntegerLiteral*](#167) .*DecimalDigits*opt*ExponentPart*opt

.*DecimalDigits ExponentPart*opt

[*DecimalIntegerLiteral*](#167)*ExponentPart*opt

[*DecimalIntegerLiteral*](#167) ::

0

[*NonZeroDigit*](#167)*DecimalDigits*opt

[*DecimalDigits*](#167) ::

[*DecimalDigit*](#167)

[*DecimalDigits*](#167)[*DecimalDigit*](#167)

[*DecimalDigit*](#167) :: one of

0 1 2 3 4 5 6 7 8 9

[*NonZeroDigit*](#167) :: one of

1 2 3 4 5 6 7 8 9

[*ExponentPart*](#167) ::

[*ExponentIndicator*](#167)[*SignedInteger*](#167)

[*ExponentIndicator*](#167) :: one of

e E

[*SignedInteger*](#167) ::

[*DecimalDigits*](#167)

+[*DecimalDigits*](#167)

‐[*DecimalDigits*](#167)

[*BinaryIntegerLiteral*](#167) ::

0b[*BinaryDigits*](#167)

0B[*BinaryDigits*](#167)

[*BinaryDigits*](#167) ::

[*BinaryDigit*](#167)

[*BinaryDigits*](#167)[*BinaryDigit*](#167)

[*BinaryDigit*](#167) :: one of

0 1

[*OctalIntegerLiteral*](#167) ::

0o[*OctalDigits*](#167)

0O[*OctalDigits*](#167)

[*OctalDigits*](#167) ::

[*OctalDigit*](#167)

[*OctalDigits*](#167)[*OctalDigit*](#167)

[*OctalDigit*](#167) :: one of

0 1 2 3 4 5 6 7

[*HexIntegerLiteral*](#168) ::

0x[*HexDigits*](#168)

0X[*HexDigits*](#168)

[*HexDigits*](#168) ::

[*HexDigit*](#168)

[*HexDigits*](#168)[*HexDigit*](#168)

[*HexDigit*](#168) :: one of

0 1 2 3 4 5 6 7 8 9 a b c d e f A B C D E F

[*StringLiteral*](#169) ::

"*DoubleStringCharacters*opt "

'*SingleStringCharacters*opt '

[*DoubleStringCharacters*](#170) ::

[*DoubleStringCharacter*](#170)*DoubleStringCharacters*opt

[*SingleStringCharacters*](#170) ::

[*SingleStringCharacter*](#170)*SingleStringCharacters*opt

[*DoubleStringCharacter*](#170) ::

[*SourceCharacter*](#158) but not one of "  or \  or [*LineTerminator*](#162)

\[*EscapeSequence*](#170)

[*LineContinuation*](#170)

[*SingleStringCharacter*](#170) ::

[*SourceCharacter*](#158) but not one of '  or \  or [*LineTerminator*](#162)

\[*EscapeSequence*](#170)

[*LineContinuation*](#170)

[*LineContinuation*](#170) ::

\[*LineTerminatorSequence*](#162)

[*EscapeSequence*](#170) ::

[*CharacterEscapeSequence*](#170)

0 [lookahead ∉ *DecimalDigit*]

[*HexEscapeSequence*](#170)

[*UnicodeEscapeSequence*](#170)

[*CharacterEscapeSequence*](#170) ::

[*SingleEscapeCharacter*](#170)

[*NonEscapeCharacter*](#170)

[*SingleEscapeCharacter*](#170) :: one of

' " \ b f n r t v

[*NonEscapeCharacter*](#170) ::

[*SourceCharacter*](#158) but not one of *EscapeCharacter*  or [*LineTerminator*](#162)

[*EscapeCharacter*](#170) ::

[*SingleEscapeCharacter*](#170)

[*DecimalDigit*](#167)

x

u

[*HexEscapeSequence*](#170) ::

x*HexDigit*[*HexDigit*](#168)

[*UnicodeEscapeSequence*](#170) ::

u[*Hex4Digits*](#170)

u{*HexDigits* }

[*Hex4Digits*](#170) ::

[*HexDigit*](#168)*HexDigit HexDigit*[*HexDigit*](#168)

[*RegularExpressionLiteral*](#172) ::

/*RegularExpressionBody* /[*RegularExpressionFlags*](#173)

[*RegularExpressionBody*](#172) ::

[*RegularExpressionFirstChar*](#173)[*RegularExpressionChars*](#172)

[*RegularExpressionChars*](#172) ::

[empty]

[*RegularExpressionChars*](#172)[*RegularExpressionChar*](#173)

[*RegularExpressionFirstChar*](#173) ::

[*RegularExpressionNonTerminator*](#173) but not one of \*  or \  or /  or [

[*RegularExpressionBackslashSequence*](#173)

[*RegularExpressionClass*](#173)

[*RegularExpressionChar*](#173) ::

[*RegularExpressionNonTerminator*](#173) but not one of \  or /  or [

[*RegularExpressionBackslashSequence*](#173)

[*RegularExpressionClass*](#173)

[*RegularExpressionBackslashSequence*](#173) ::

\[*RegularExpressionNonTerminator*](#173)

[*RegularExpressionNonTerminator*](#173) ::

[*SourceCharacter*](#158) but not [*LineTerminator*](#162)

[*RegularExpressionClass*](#173) ::

[*RegularExpressionClassChars* ]

[*RegularExpressionClassChars*](#173) ::

[empty]

[*RegularExpressionClassChars*](#173)[*RegularExpressionClassChar*](#173)

[*RegularExpressionClassChar*](#173) ::

[*RegularExpressionNonTerminator*](#173) but not one of ]  or \

[*RegularExpressionBackslashSequence*](#173)

[*RegularExpressionFlags*](#173) ::

[empty]

[*RegularExpressionFlags*](#173)[*IdentiᲪierPart*](#164)

[*Template*](#173) ::

[*NoSubstitutionTemplate*](#174)

[*TemplateHead*](#174)

[*NoSubstitutionTemplate*](#174) ::

`*TemplateCharacters*opt `

[*TemplateHead*](#174) ::

`*TemplateCharacters*opt ${

[*TemplateSubstitutionTail*](#174) ::

[*TemplateMiddle*](#174)

[*TemplateTail*](#174)

[*TemplateMiddle*](#174) ::

}*TemplateCharacters*opt ${

[*TemplateTail*](#174) ::

}*TemplateCharacters*opt `

[*TemplateCharacters*](#174) ::

[*TemplateCharacter*](#174)*TemplateCharacters*opt

[*TemplateCharacter*](#174) ::

$ [lookahead ≠ {]

\[*EscapeSequence*](#170)

[*LineContinuation*](#170)

[*LineTerminatorSequence*](#162)

[*SourceCharacter*](#158) but not one of `  or \  or $  or [*LineTerminator*](#162)

A.2 Expressions

[*IdentiᲪierReference*](#178)[Yield] :

[*IdentiᲪier*](#178)

[~Yield] yield

[*BindingIdentiᲪier*](#178)[Yield] :

[*IdentiᲪier*](#178)

[~Yield] yield

[*IdentiᲪier*](#178) :

[*IdentiᲪierName*](#164) but not [*ReservedWord*](#165)

[*LabelIdentiᲪier*](#178)[Yield] :

[*IdentiᲪier*](#178)

[~Yield] yield

[*PrimaryExpression*](#180)[Yield] :

this

[*IdentiᲪierReference*](#178)[?Yield]

[*Literal*](#182)

[*ArrayLiteral*](#183)[?Yield]

[*ObjectLiteral*](#185)[?Yield]

[*FunctionExpression*](#275)

[*ClassExpression*](#294)[?Yield]

[*GeneratorExpression*](#289)

[*RegularExpressionLiteral*](#172)

[*TemplateLiteral*](#189)[?Yield]

[*CoverParenthesizedExpressionAndArrowParameterList*](#181)[?Yield]

[*CoverParenthesizedExpressionAndArrowParameterList*](#181)[Yield] :

(*Expression*[In, ?Yield] )

( )

( ... *BindingIdentiᲪier*[?Yield] )

( ... *BindingPattern*[?Yield] )

(*Expression*[In, ?Yield] , ... *BindingIdentiᲪier*[?Yield] )

(*Expression*[In, ?Yield] , ... *BindingPattern*[?Yield] )

When processing the production *PrimaryExpression* :*CoverParenthesizedExpressionAndArrowParameterList*  the

interpretation of *CoverParenthesizedExpressionAndArrowParameterList* is re伀氂ined using the following grammar:

[*ParenthesizedExpression*](#181)[Yield] :

(*Expression*[In, ?Yield] )

[*Literal*](#182) :

[*NullLiteral*](#166)

[*BooleanLiteral*](#166)

[*NumericLiteral*](#167)

[*StringLiteral*](#169)

[*ArrayLiteral*](#183)[Yield] :

[*Elision*opt ]

[*ElementList*[?Yield] ]

[*ElementList*[?Yield] ,*Elision*opt ]

[*ElementList*](#183)[Yield] :

[*Elision*](#183)opt*AssignmentExpression*[In, ?Yield]

[*Elision*](#183)opt*SpreadElement*[?Yield]

[*ElementList*](#183)[?Yield] ,*Elision*opt*AssignmentExpression*[In, ?Yield]

[*ElementList*](#183)[?Yield] ,*Elision*opt*SpreadElement*[?Yield]

[*Elision*](#183) :

,

[*Elision*](#183) ,

[*SpreadElement*](#183)[Yield] :

... *AssignmentExpression*[In, ?Yield]

[*ObjectLiteral*](#185)[Yield] :

{ }

{*PropertyDeᲪinitionList*[?Yield] }

{*PropertyDeᲪinitionList*[?Yield] , }

[*PropertyDeᲪinitionList*](#185)[Yield] :

[*PropertyDeᲪinition*](#185)[?Yield]

[*PropertyDeᲪinitionList*](#185)[?Yield] ,*PropertyDeᲪinition*[?Yield]

[*PropertyDeᲪinition*](#185)[Yield] :

[*IdentiᲪierReference*](#178)[?Yield]

[*CoverInitializedName*](#186)[?Yield]

[*PropertyName*](#185)[?Yield] :*AssignmentExpression*[In, ?Yield]

[*MethodDeᲪinition*](#286)[?Yield]

[*PropertyName*](#185)[Yield] :

[*LiteralPropertyName*](#185)

[*ComputedPropertyName*](#186)[?Yield]

[*LiteralPropertyName*](#185) :

[*IdentiᲪierName*](#164)

[*StringLiteral*](#169)

[*NumericLiteral*](#167)

[*ComputedPropertyName*](#186)[Yield] :

[*AssignmentExpression*[In, ?Yield] ]

[*CoverInitializedName*](#186)[Yield] :

[*IdentiᲪierReference*](#178)[?Yield]*Initializer*[In, ?Yield]

[*Initializer*](#186)[In, Yield] :

=*AssignmentExpression*[?In, ?Yield]

[*TemplateLiteral*](#189)[Yield] :

[*NoSubstitutionTemplate*](#174)

[*TemplateHead*](#174)*Expression*[In, ?Yield]*TemplateSpans*[?Yield]

[*TemplateSpans*](#189)[Yield] :

[*TemplateTail*](#174)

[*TemplateMiddleList*](#189)[?Yield][*TemplateTail*](#174)

[*TemplateMiddleList*](#189)[Yield] :

[*TemplateMiddle*](#174)*Expression*[In, ?Yield]

[*TemplateMiddleList*](#189)[?Yield]*TemplateMiddle Expression*[In, ?Yield]

[*MemberExpression*](#194)[Yield] :

[*PrimaryExpression*](#180)[?Yield]

[*MemberExpression*](#194)[?Yield] [*Expression*[In, ?Yield] ]

[*MemberExpression*](#194)[?Yield] .[*IdentiᲪierName*](#164)

[*MemberExpression*](#194)[?Yield]*TemplateLiteral*[?Yield]

[*SuperProperty*](#194)[?Yield]

[*MetaProperty*](#194)

new*MemberExpression*[?Yield]*Arguments*[?Yield]

[*SuperProperty*](#194)[Yield] :

super [*Expression*[In, ?Yield] ]

super .[*IdentiᲪierName*](#164)

[*MetaProperty*](#194) :

[*NewTarget*](#194)

[*NewTarget*](#194) :

new . target

[*NewExpression*](#194)[Yield] :

[*MemberExpression*](#194)[?Yield]

new*NewExpression*[?Yield]

[*CallExpression*](#194)[Yield] :

[*MemberExpression*](#194)[?Yield]*Arguments*[?Yield]

[*SuperCall*](#194)[?Yield]

[*CallExpression*](#194)[?Yield]*Arguments*[?Yield]

[*CallExpression*](#194)[?Yield] [*Expression*[In, ?Yield] ]

[*CallExpression*](#194)[?Yield] .[*IdentiᲪierName*](#164)

[*CallExpression*](#194)[?Yield]*TemplateLiteral*[?Yield]

[*SuperCall*](#194)[Yield] :

super*Arguments*[?Yield]

[*Arguments*](#194)[Yield] :

( )

(*ArgumentList*[?Yield] )

[*ArgumentList*](#194)[Yield] :

[*AssignmentExpression*](#218)[In, ?Yield]

... *AssignmentExpression*[In, ?Yield]

[*ArgumentList*](#194)[?Yield] ,*AssignmentExpression*[In, ?Yield]

[*ArgumentList*](#194)[?Yield] , ... *AssignmentExpression*[In, ?Yield]

[*LeftHandSideExpression*](#194)[Yield] :

[*NewExpression*](#194)[?Yield]

[*CallExpression*](#194)[?Yield]

[*UpdateExpression*](#202)[Yield] :

[*LeftHandSideExpression*](#194)[?Yield]

[*LeftHandSideExpression*](#194)[?Yield] [no *LineTerminator* here] ++

[*LeftHandSideExpression*](#194)[?Yield] [no *LineTerminator* here] ‐‐

++*UnaryExpression*[?Yield]

‐‐*UnaryExpression*[?Yield]

[*UnaryExpression*](#203)[Yield] :

[*UpdateExpression*](#202)[?Yield]

delete*UnaryExpression*[?Yield]

void*UnaryExpression*[?Yield]

typeof*UnaryExpression*[?Yield]

+*UnaryExpression*[?Yield]

‐*UnaryExpression*[?Yield]

~*UnaryExpression*[?Yield]

!*UnaryExpression*[?Yield]

[*ExponentiationExpression*](#206)[Yield] :

[*UnaryExpression*](#203)[?Yield]

[*UpdateExpression*](#202)[?Yield] \*\**ExponentiationExpression*[?Yield]

[*MultiplicativeExpression*](#207)[Yield] :

[*ExponentiationExpression*](#206)[?Yield]

[*MultiplicativeExpression*](#207)[?Yield]*MultiplicativeOperator ExponentiationExpression*[?Yield]

[*MultiplicativeOperator*](#207) : one of

\* / %

[*AdditiveExpression*](#209)[Yield] :

[*MultiplicativeExpression*](#207)[?Yield]

[*AdditiveExpression*](#209)[?Yield] +*MultiplicativeExpression*[?Yield]

[*AdditiveExpression*](#209)[?Yield] ‐*MultiplicativeExpression*[?Yield]

[*ShiftExpression*](#211)[Yield] :

[*AdditiveExpression*](#209)[?Yield]

[*ShiftExpression*](#211)[?Yield] <<*AdditiveExpression*[?Yield]

[*ShiftExpression*](#211)[?Yield] >>*AdditiveExpression*[?Yield]

[*ShiftExpression*](#211)[?Yield] >>>*AdditiveExpression*[?Yield]

[*RelationalExpression*](#212)[In, Yield] :

[*ShiftExpression*](#211)[?Yield]

[*RelationalExpression*](#212)[?In, ?Yield] <*ShiftExpression*[?Yield]

[*RelationalExpression*](#212)[?In, ?Yield] >*ShiftExpression*[?Yield]

[*RelationalExpression*](#212)[?In, ?Yield] <=*ShiftExpression*[?Yield]

[*RelationalExpression*](#212)[?In, ?Yield] >=*ShiftExpression*[?Yield]

[*RelationalExpression*](#212)[?In, ?Yield] instanceof*ShiftExpression*[?Yield]

[+In] [*RelationalExpression*](#212)[In, ?Yield] in*ShiftExpression*[?Yield]

[*EqualityExpression*](#215)[In, Yield] :

[*RelationalExpression*](#212)[?In, ?Yield]

[*EqualityExpression*](#215)[?In, ?Yield] ==*RelationalExpression*[?In, ?Yield]

[*EqualityExpression*](#215)[?In, ?Yield] !=*RelationalExpression*[?In, ?Yield]

[*EqualityExpression*](#215)[?In, ?Yield] ===*RelationalExpression*[?In, ?Yield]

[*EqualityExpression*](#215)[?In, ?Yield] !==*RelationalExpression*[?In, ?Yield]

[*BitwiseANDExpression*](#216)[In, Yield] :

[*EqualityExpression*](#215)[?In, ?Yield]

[*BitwiseANDExpression*](#216)[?In, ?Yield] &*EqualityExpression*[?In, ?Yield]

[*BitwiseXORExpression*](#216)[In, Yield] :

[*BitwiseANDExpression*](#216)[?In, ?Yield]

[*BitwiseXORExpression*](#216)[?In, ?Yield] ^*BitwiseANDExpression*[?In, ?Yield]

[*BitwiseORExpression*](#216)[In, Yield] :

[*BitwiseXORExpression*](#216)[?In, ?Yield]

[*BitwiseORExpression*](#216)[?In, ?Yield] |*BitwiseXORExpression*[?In, ?Yield]

[*LogicalANDExpression*](#217)[In, Yield] :

[*BitwiseORExpression*](#216)[?In, ?Yield]

[*LogicalANDExpression*](#217)[?In, ?Yield] &&*BitwiseORExpression*[?In, ?Yield]

[*LogicalORExpression*](#217)[In, Yield] :

[*LogicalANDExpression*](#217)[?In, ?Yield]

[*LogicalORExpression*](#217)[?In, ?Yield] ||*LogicalANDExpression*[?In, ?Yield]

[*ConditionalExpression*](#218)[In, Yield] :

[*LogicalORExpression*](#217)[?In, ?Yield]

[*LogicalORExpression*](#217)[?In, ?Yield] ?*AssignmentExpression*[In, ?Yield] :*AssignmentExpression*[?In, ?Yield]

[*AssignmentExpression*](#218)[In, Yield] :

[*ConditionalExpression*](#218)[?In, ?Yield]

[+Yield] [*YieldExpression*](#289)[?In]

[*ArrowFunction*](#283)[?In, ?Yield]

[*LeftHandSideExpression*](#194)[?Yield] =*AssignmentExpression*[?In, ?Yield]

[*LeftHandSideExpression*](#194)[?Yield]*AssignmentOperator AssignmentExpression*[?In, ?Yield]

In certain circumstances when processing the production *AssignmentExpression* :*LeftHandSideExpression* =

[*AssignmentExpression*](#218)  the following grammar is used to re伀氂ine the interpretation of [*LeftHandSideExpression*:](#194)

[*AssignmentPattern*](#220)[Yield] :

[*ObjectAssignmentPattern*](#220)[?Yield]

[*ArrayAssignmentPattern*](#220)[?Yield]

[*ObjectAssignmentPattern*](#220)[Yield] :

{ }

{*AssignmentPropertyList*[?Yield] }

{*AssignmentPropertyList*[?Yield] , }

[*ArrayAssignmentPattern*](#220)[Yield] :

[*Elision*opt*AssignmentRestElement*[?Yield] opt ]

[*AssignmentElementList*[?Yield] ]

[*AssignmentElementList*[?Yield] ,*Elision*opt*AssignmentRestElement*[?Yield] opt ]

[*AssignmentPropertyList*](#220)[Yield] :

[*AssignmentProperty*](#221)[?Yield]

[*AssignmentPropertyList*](#220)[?Yield] ,*AssignmentProperty*[?Yield]

[*AssignmentElementList*](#220)[Yield] :

[*AssignmentElisionElement*](#221)[?Yield]

[*AssignmentElementList*](#220)[?Yield] ,*AssignmentElisionElement*[?Yield]

[*AssignmentElisionElement*](#221)[Yield] :

[*Elision*](#183)opt*AssignmentElement*[?Yield]

[*AssignmentProperty*](#221)[Yield] :

[*IdentiᲪierReference*](#178)[?Yield]*Initializer*[In, ?Yield] opt

[*PropertyName*](#185)[?Yield] :*AssignmentElement*[?Yield]

[*AssignmentElement*](#221)[Yield] :

[*DestructuringAssignmentTarget*](#221)[?Yield]*Initializer*[In, ?Yield] opt

[*AssignmentRestElement*](#221)[Yield] :

... *DestructuringAssignmentTarget*[?Yield]

[*DestructuringAssignmentTarget*](#221)[Yield] :

[*LeftHandSideExpression*](#194)[?Yield]

[*AssignmentOperator*](#219) : one of

\*= /= %= += ‐= <<= >>= >>>= &= ^= |= \*\*=

[*Expression*](#225)[In, Yield] :

[*AssignmentExpression*](#218)[?In, ?Yield]

[*Expression*](#225)[?In, ?Yield] ,*AssignmentExpression*[?In, ?Yield]

A.3 Statements

[*Statement*](#226)[Yield, Return] :

[*BlockStatement*](#229)[?Yield, ?Return]

[*VariableStatement*](#236)[?Yield]

[*EmptyStatement*](#245)

[*ExpressionStatement*](#245)[?Yield]

[*IfStatement*](#245)[?Yield, ?Return]

[*BreakableStatement*](#226)[?Yield, ?Return]

[*ContinueStatement*](#258)[?Yield]

[*BreakStatement*](#259)[?Yield]

[+Return] [*ReturnStatement*](#260)[?Yield]

[*WithStatement*](#260)[?Yield, ?Return]

[*LabelledStatement*](#268)[?Yield, ?Return]

[*ThrowStatement*](#271)[?Yield]

[*TryStatement*](#271)[?Yield, ?Return]

[*DebuggerStatement*](#275)

[*Declaration*](#226)[Yield] :

[*HoistableDeclaration*](#226)[?Yield]

[*ClassDeclaration*](#294)[?Yield]

[*LexicalDeclaration*](#234)[In, ?Yield]

[*HoistableDeclaration*](#226)[Yield, Default] :

[*FunctionDeclaration*](#275)[?Yield, ?Default]

[*GeneratorDeclaration*](#289)[?Yield, ?Default]

[*BreakableStatement*](#226)[Yield, Return] :

[*IterationStatement*](#247)[?Yield, ?Return]

[*SwitchStatement*](#261)[?Yield, ?Return]

[*BlockStatement*](#229)[Yield, Return] :

[*Block*](#54)[?Yield, ?Return]

[*Block*](#54) :

{*StatementList* }

[*StatementList*](#229)[Yield, Return] :

[*StatementListItem*](#229)[?Yield, ?Return]

[*StatementList*](#229)[?Yield, ?Return]*StatementListItem*[?Yield, ?Return]

[*StatementListItem*](#229)[Yield, Return] :

[*Statement*](#226)[?Yield, ?Return]

[*Declaration*](#226)[?Yield]

[*LexicalDeclaration*](#234)[In, Yield] :

[*LetOrConst*](#234)*BindingList*[?In, ?Yield] ;

[*LetOrConst*](#234) :

let

const

[*BindingList*](#234)[In, Yield] :

[*LexicalBinding*](#234)[?In, ?Yield]

[*BindingList*](#234)[?In, ?Yield] ,*LexicalBinding*[?In, ?Yield]

[*LexicalBinding*](#234)[In, Yield] :

[*BindingIdentiᲪier*](#178)[?Yield]*Initializer*[?In, ?Yield] opt

[*BindingPattern*](#237)[?Yield]*Initializer*[?In, ?Yield]

[*VariableStatement*](#236)[Yield] :

var*VariableDeclarationList*[In, ?Yield] ;

[*VariableDeclarationList*](#236)[In, Yield] :

[*VariableDeclaration*](#236)[?In, ?Yield]

[*VariableDeclarationList*](#236)[?In, ?Yield] ,*VariableDeclaration*[?In, ?Yield]

[*VariableDeclaration*](#236)[In, Yield] :

[*BindingIdentiᲪier*](#178)[?Yield]*Initializer*[?In, ?Yield] opt

[*BindingPattern*](#237)[?Yield]*Initializer*[?In, ?Yield]

[*BindingPattern*](#237)[Yield] :

[*ObjectBindingPattern*](#237)[?Yield]

[*ArrayBindingPattern*](#237)[?Yield]

[*ObjectBindingPattern*](#237)[Yield] :

{ }

{*BindingPropertyList*[?Yield] }

{*BindingPropertyList*[?Yield] , }

[*ArrayBindingPattern*](#237)[Yield] :

[*Elision*opt*BindingRestElement*[?Yield] opt ]

[*BindingElementList*[?Yield] ]

[*BindingElementList*[?Yield] ,*Elision*opt*BindingRestElement*[?Yield] opt ]

[*BindingPropertyList*](#238)[Yield] :

[*BindingProperty*](#238)[?Yield]

[*BindingPropertyList*](#238)[?Yield] ,*BindingProperty*[?Yield]

[*BindingElementList*](#238)[Yield] :

[*BindingElisionElement*](#238)[?Yield]

[*BindingElementList*](#238)[?Yield] ,*BindingElisionElement*[?Yield]

[*BindingElisionElement*](#238)[Yield] :

[*Elision*](#183)opt*BindingElement*[?Yield]

[*BindingProperty*](#238)[Yield] :

[*SingleNameBinding*](#238)[?Yield]

[*PropertyName*](#185)[?Yield] :*BindingElement*[?Yield]

[*BindingElement*](#238)[Yield] :

[*SingleNameBinding*](#238)[?Yield]

[*BindingPattern*](#237)[?Yield]*Initializer*[In, ?Yield] opt

[*SingleNameBinding*](#238)[Yield] :

[*BindingIdentiᲪier*](#178)[?Yield]*Initializer*[In, ?Yield] opt

[*BindingRestElement*](#238)[Yield] :

... *BindingIdentiᲪier*[?Yield]

... *BindingPattern*[?Yield]

[*EmptyStatement*](#245) :

;

[*ExpressionStatement*](#245)[Yield] :

[lookahead ∉ { { , function , class , let [ }]*Expression*[In, ?Yield] ;

[*IfStatement*](#245)[Yield, Return] :

if (*Expression*[In, ?Yield] )*Statement*[?Yield, ?Return] else*Statement*[?Yield, ?Return]

if (*Expression*[In, ?Yield] )*Statement*[?Yield, ?Return]

[*IterationStatement*](#247)[Yield, Return] :

do*Statement*[?Yield, ?Return] while (*Expression*[In, ?Yield] ) ;

while (*Expression*[In, ?Yield] )*Statement*[?Yield, ?Return]

for ( [lookahead ∉ { let [ }]*Expression*[?Yield] opt ;*Expression*[In, ?Yield] opt ;

[*Expression*](#225)[In, ?Yield] opt )*Statement*[?Yield, ?Return]

for ( var*VariableDeclarationList*[?Yield] ;*Expression*[In, ?Yield] opt ;*Expression*[In, ?Yield] opt )

[*Statement*](#226)[?Yield, ?Return]

for (*LexicalDeclaration*[?Yield]*Expression*[In, ?Yield] opt ;*Expression*[In, ?Yield] opt )

[*Statement*](#226)[?Yield, ?Return]

for ( [lookahead ∉ { let [ }]*LeftHandSideExpression*[?Yield] in*Expression*[In, ?Yield] )

[*Statement*](#226)[?Yield, ?Return]

for ( var*ForBinding*[?Yield] in*Expression*[In, ?Yield] )*Statement*[?Yield, ?Return]

for (*ForDeclaration*[?Yield] in*Expression*[In, ?Yield] )*Statement*[?Yield, ?Return]

for ( [lookahead ≠ let]*LeftHandSideExpression*[?Yield] of*AssignmentExpression*[In, ?Yield] )

[*Statement*](#226)[?Yield, ?Return]

for ( var*ForBinding*[?Yield] of*AssignmentExpression*[In, ?Yield] )*Statement*[?Yield, ?Return]

for (*ForDeclaration*[?Yield] of*AssignmentExpression*[In, ?Yield] )*Statement*[?Yield, ?Return]

[*ForDeclaration*](#247)[Yield] :

[*LetOrConst*](#234)*ForBinding*[?Yield]

[*ForBinding*](#248)[Yield] :

[*BindingIdentiᲪier*](#178)[?Yield]

[*BindingPattern*](#237)[?Yield]

[*ContinueStatement*](#258)[Yield] :

continue ;

continue [no *LineTerminator* here]*LabelIdentiᲪier*[?Yield] ;

[*BreakStatement*](#259)[Yield] :

break ;

break [no *LineTerminator* here]*LabelIdentiᲪier*[?Yield] ;

[*ReturnStatement*](#260)[Yield] :

return ;

return [no *LineTerminator* here]*Expression*[In, ?Yield] ;

[*WithStatement*](#260)[Yield, Return] :

with (*Expression*[In, ?Yield] )*Statement*[?Yield, ?Return]

[*SwitchStatement*](#261)[Yield, Return] :

switch (*Expression*[In, ?Yield] )*CaseBlock*[?Yield, ?Return]

[*CaseBlock*](#261)[Yield, Return] :

{*CaseClauses*[?Yield, ?Return] opt }

{*CaseClauses*[?Yield, ?Return] opt*DefaultClause*[?Yield, ?Return]*CaseClauses*[?Yield, ?Return] opt }

[*CaseClauses*](#262)[Yield, Return] :

[*CaseClause*](#262)[?Yield, ?Return]

[*CaseClauses*](#262)[?Yield, ?Return]*CaseClause*[?Yield, ?Return]

[*CaseClause*](#262)[Yield, Return] :

case*Expression*[In, ?Yield] :*StatementList*[?Yield, ?Return] opt

[*DefaultClause*](#262)[Yield, Return] :

default :*StatementList*[?Yield, ?Return] opt

[*LabelledStatement*](#268)[Yield, Return] :

[*LabelIdentiᲪier*](#178)[?Yield] :*LabelledItem*[?Yield, ?Return]

[*LabelledItem*](#268)[Yield, Return] :

[*Statement*](#226)[?Yield, ?Return]

[*FunctionDeclaration*](#275)[?Yield]

[*ThrowStatement*](#271)[Yield] :

throw [no *LineTerminator* here]*Expression*[In, ?Yield] ;

[*TryStatement*](#271)[Yield, Return] :

try*Block*[?Yield, ?Return]*Catch*[?Yield, ?Return]

try*Block*[?Yield, ?Return]*Finally*[?Yield, ?Return]

try*Block*[?Yield, ?Return]*Catch*[?Yield, ?Return]*Finally*[?Yield, ?Return]

[*Catch*](#271)[Yield, Return] :

catch (*CatchParameter*[?Yield] )*Block*[?Yield, ?Return]

[*Finally*](#271)[Yield, Return] :

finally*Block*[?Yield, ?Return]

[*CatchParameter*](#271)[Yield] :

[*BindingIdentiᲪier*](#178)[?Yield]

[*BindingPattern*](#237)[?Yield]

[*DebuggerStatement*](#275) :

debugger ;

A.4 Functions and Classes

[*FunctionDeclaration*](#275)[Yield, Default] :

function*BindingIdentiᲪier*[?Yield] (*FormalParameters* ) {*FunctionBody* }

[+Default] function (*FormalParameters* ) {*FunctionBody* }

[*FunctionExpression*](#275) :

function*BindingIdentiᲪier*opt (*FormalParameters* ) {*FunctionBody* }

[*StrictFormalParameters*](#275)[Yield] :

[*FormalParameters*](#276)[?Yield]

[*FormalParameters*](#276)[Yield] :

[empty]

[*FormalParameterList*](#276)[?Yield]

[*FormalParameterList*](#276)[Yield] :

[*FunctionRestParameter*](#276)[?Yield]

[*FormalsList*](#276)[?Yield]

[*FormalsList*](#276)[?Yield] ,*FunctionRestParameter*[?Yield]

[*FormalsList*](#276)[Yield] :

[*FormalParameter*](#276)[?Yield]

[*FormalsList*](#276)[?Yield] ,*FormalParameter*[?Yield]

[*FunctionRestParameter*](#276)[Yield] :

[*BindingRestElement*](#238)[?Yield]

[*FormalParameter*](#276)[Yield] :

[*BindingElement*](#238)[?Yield]

[*FunctionBody*](#276)[Yield] :

[*FunctionStatementList*](#276)[?Yield]

[*FunctionStatementList*](#276)[Yield] :

[*StatementList*](#229)[?Yield, Return] opt

[*ArrowFunction*](#283)[In, Yield] :

[*ArrowParameters*](#283)[?Yield] [no *LineTerminator* here] =>*ConciseBody*[?In]

[*ArrowParameters*](#283)[Yield] :

[*BindingIdentiᲪier*](#178)[?Yield]

[*CoverParenthesizedExpressionAndArrowParameterList*](#181)[?Yield]

[*ConciseBody*](#283)[In] :

[lookahead ≠ {]*AssignmentExpression*[?In]

{*FunctionBody* }

When the production *ArrowParameters* :*CoverParenthesizedExpressionAndArrowParameterList*  is recognized the following

grammar is used to re伀氂ine the interpretation of [*CoverParenthesizedExpressionAndArrowParameterList*:](#181)

[*ArrowFormalParameters*](#283)[Yield] :

(*StrictFormalParameters*[?Yield] )

[*MethodDeᲪinition*](#286)[Yield] :

[*PropertyName*](#185)[?Yield] (*StrictFormalParameters* ) {*FunctionBody* }

[*GeneratorMethod*](#289)[?Yield]

get*PropertyName*[?Yield] ( ) {*FunctionBody* }

set*PropertyName*[?Yield] (*PropertySetParameterList* ) {*FunctionBody* }

[*PropertySetParameterList*](#286) :

[*FormalParameter*](#276)

[*GeneratorMethod*](#289)[Yield] :

\**PropertyName*[?Yield] (*StrictFormalParameters*[Yield] ) {*GeneratorBody* }

[*GeneratorDeclaration*](#289)[Yield, Default] :

function \**BindingIdentiᲪier*[?Yield] (*FormalParameters*[Yield] ) {*GeneratorBody* }

[+Default] function \* (*FormalParameters*[Yield] ) {*GeneratorBody* }

[*GeneratorExpression*](#289) :

function \**BindingIdentiᲪier*[Yield] opt (*FormalParameters*[Yield] ) {*GeneratorBody* }

[*GeneratorBody*](#289) :

[*FunctionBody*](#276)[Yield]

[*YieldExpression*](#289)[In] :

yield

yield [no *LineTerminator* here]*AssignmentExpression*[?In, Yield]

yield [no *LineTerminator* here] \**AssignmentExpression*[?In, Yield]

[*ClassDeclaration*](#294)[Yield, Default] :

class*BindingIdentiᲪier*[?Yield]*ClassTail*[?Yield]

[+Default] class*ClassTail*[?Yield]

[*ClassExpression*](#294)[Yield] :

class*BindingIdentiᲪier*[?Yield] opt*ClassTail*[?Yield]

[*ClassTail*](#294)[Yield] :

[*ClassHeritage*](#294)[?Yield] opt {*ClassBody*[?Yield] opt }

[*ClassHeritage*](#294)[Yield] :

extends*LeftHandSideExpression*[?Yield]

[*ClassBody*](#294)[Yield] :

[*ClassElementList*](#294)[?Yield]

[*ClassElementList*](#294)[Yield] :

[*ClassElement*](#294)[?Yield]

[*ClassElementList*](#294)[?Yield]*ClassElement*[?Yield]

[*ClassElement*](#294)[Yield] :

[*MethodDeᲪinition*](#286)[?Yield]

static*MethodDeᲪinition*[?Yield]

;

A.5 Scripts and Modules

[*Script*](#304) :

[*ScriptBody*](#304)opt

[*ScriptBody*](#304) :

[*StatementList*](#229)

[*Module*](#308) :

[*ModuleBody*](#308)opt

[*ModuleBody*](#308) :

[*ModuleItemList*](#308)

[*ModuleItemList*](#308) :

[*ModuleItem*](#308)

[*ModuleItemList*](#308)[*ModuleItem*](#308)

[*ModuleItem*](#308) :

[*ImportDeclaration*](#320)

[*ExportDeclaration*](#323)

[*StatementListItem*](#229)

[*ImportDeclaration*](#320) :

import*ImportClause FromClause* ;

import*ModuleSpeciᲪier* ;

[*ImportClause*](#321) :

[*ImportedDefaultBinding*](#321)

[*NameSpaceImport*](#321)

[*NamedImports*](#321)

[*ImportedDefaultBinding*](#321) ,[*NameSpaceImport*](#321)

[*ImportedDefaultBinding*](#321) ,[*NamedImports*](#321)

[*ImportedDefaultBinding*](#321) :

[*ImportedBinding*](#321)

[*NameSpaceImport*](#321) :

\* as[*ImportedBinding*](#321)

[*NamedImports*](#321) :

{ }

{*ImportsList* }

{*ImportsList* , }

[*FromClause*](#321) :

from[*ModuleSpeciᲪier*](#321)

[*ImportsList*](#321) :

[*ImportSpeciᲪier*](#321)

[*ImportsList*](#321) ,[*ImportSpeciᲪier*](#321)

[*ImportSpeciᲪier*](#321) :

[*ImportedBinding*](#321)

[*IdentiᲪierName*](#164) as[*ImportedBinding*](#321)

[*ModuleSpeciᲪier*](#321) :

[*StringLiteral*](#169)

[*ImportedBinding*](#321) :

[*BindingIdentiᲪier*](#178)

[*ExportDeclaration*](#323) :

export \**FromClause* ;

export*ExportClause FromClause* ;

export*ExportClause* ;

export[*VariableStatement*](#236)

export[*Declaration*](#226)

export default*HoistableDeclaration*[Default]

export default*ClassDeclaration*[Default]

export default [lookahead ∉ { function , class }]*AssignmentExpression*[In] ;

[*ExportClause*](#323) :

{ }

{*ExportsList* }

{*ExportsList* , }

[*ExportsList*](#323) :

[*ExportSpeciᲪier*](#323)

[*ExportsList*](#323) ,[*ExportSpeciᲪier*](#323)

[*ExportSpeciᲪier*](#323) :

[*IdentiᲪierName*](#164)

[*IdentiᲪierName*](#164) as[*IdentiᲪierName*](#164)

A.6 Number Conversions

[*StringNumericLiteral*](#78) :::

[*StrWhiteSpace*](#78)opt

[*StrWhiteSpace*](#78)opt*StrNumericLiteral StrWhiteSpace*opt

[*StrWhiteSpace*](#78) :::

[*StrWhiteSpaceChar*](#78)*StrWhiteSpace*opt

[*StrWhiteSpaceChar*](#78) :::

[*WhiteSpace*](#162)

[*LineTerminator*](#162)

[*StrNumericLiteral*](#78) :::

[*StrDecimalLiteral*](#78)

[*BinaryIntegerLiteral*](#167)

[*OctalIntegerLiteral*](#167)

[*HexIntegerLiteral*](#168)

[*StrDecimalLiteral*](#78) :::

[*StrUnsignedDecimalLiteral*](#78)

+[*StrUnsignedDecimalLiteral*](#78)

‐[*StrUnsignedDecimalLiteral*](#78)

[*StrUnsignedDecimalLiteral*](#78) :::

Infinity

[*DecimalDigits*](#167) .*DecimalDigits*opt*ExponentPart*opt

.*DecimalDigits ExponentPart*opt

[*DecimalDigits*](#167)*ExponentPart*opt

[*DecimalDigits*](#167) ::

[*DecimalDigit*](#167)

[*DecimalDigits*](#167)[*DecimalDigit*](#167)

[*DecimalDigit*](#167) :: one of

0 1 2 3 4 5 6 7 8 9

[*ExponentPart*](#167) ::

[*ExponentIndicator*](#167)[*SignedInteger*](#167)

[*ExponentIndicator*](#167) :: one of

e E

[*SignedInteger*](#167) ::

[*DecimalDigits*](#167)

+[*DecimalDigits*](#167)

‐[*DecimalDigits*](#167)

[*HexIntegerLiteral*](#168) ::

0x[*HexDigits*](#168)

0X[*HexDigits*](#168)

[*HexDigit*](#168) :: one of

0 1 2 3 4 5 6 7 8 9 a b c d e f A B C D E F

[All grammar symbols not explicitly de伀氂ined by the *StringNumericLiteral* grammar have the de伀氂initions used in the Lexical](#166)

[Grammar for numeric literals.](#166)

A.7 Universal Resource IdentiᲪier Character Classes

[*uri*](#337) :::

[*uriCharacters*](#337)opt

[*uriCharacters*](#337) :::

[*uriCharacter*](#337)*uriCharacters*opt

[*uriCharacter*](#337) :::

[*uriReserved*](#337)

[*uriUnescaped*](#337)

[*uriEscaped*](#337)

[*uriReserved*](#337) ::: one of

; / ? : @ & = + $ ,

[*uriUnescaped*](#337) :::

[*uriAlpha*](#337)

[*DecimalDigit*](#167)

[*uriMark*](#338)

[*uriEscaped*](#337) :::

%*HexDigit*[*HexDigit*](#168)

[*uriAlpha*](#337) ::: one of

a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V

W X Y Z

[*uriMark*](#338) ::: one of

‐ \_ . ! ~ \* ' ( )

A.8 Regular Expressions

[*Pattern*](#409)[U] ::

[*Disjunction*](#409)[?U]

[*Disjunction*](#409)[U] ::

[*Alternative*](#409)[?U]

[*Alternative*](#409)[?U] |*Disjunction*[?U]

[*Alternative*](#409)[U] ::

[empty]

[*Alternative*](#409)[?U]*Term*[?U]

[*Term*](#409)[U] ::

[*Assertion*](#409)[?U]

[*Atom*](#410)[?U]

[*Atom*](#410)[?U][*QuantiᲪier*](#409)

[*Assertion*](#409)[U] ::

^

$

\ b

\ B

( ? =*Disjunction*[?U] )

( ? !*Disjunction*[?U] )

[*QuantiᲪier*](#409) ::

[*QuantiᲪierPreᲪix*](#409)

[*QuantiᲪierPreᲪix*](#409) ?

[*QuantiᲪierPreᲪix*](#409) ::

\*

+

?

{*DecimalDigits* }

{*DecimalDigits* , }

{*DecimalDigits* ,*DecimalDigits* }

[*Atom*](#410)[U] ::

[*PatternCharacter*](#410)

.

\*AtomEscape*[?U]

[*CharacterClass*](#411)[?U]

(*Disjunction*[?U] )

( ? :*Disjunction*[?U] )

[*SyntaxCharacter*](#410) :: one of

^ $ \ . \* + ? ( ) [ ] { } |

[*PatternCharacter*](#410) ::

[*SourceCharacter*](#158) but not [*SyntaxCharacter*](#410)

[*AtomEscape*](#410)[U] ::

[*DecimalEscape*](#411)

[*CharacterEscape*](#410)[?U]

[*CharacterClassEscape*](#411)

[*CharacterEscape*](#410)[U] ::

[*ControlEscape*](#410)

c[*ControlLetter*](#410)

[*HexEscapeSequence*](#170)

[*RegExpUnicodeEscapeSequence*](#410)[?U]

[*IdentityEscape*](#411)[?U]

[*ControlEscape*](#410) :: one of

f n r t v

[*ControlLetter*](#410) :: one of

a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V

W X Y Z

[*RegExpUnicodeEscapeSequence*](#410)[U] ::

[+U] u*LeadSurrogate* \u[*TrailSurrogate*](#410)

[+U] u[*LeadSurrogate*](#410)

[+U] u[*TrailSurrogate*](#410)

[+U] u[*NonSurrogate*](#410)

[~U] u[*Hex4Digits*](#170)

[+U] u{*HexDigits* }

Each \u *TrailSurrogate* for which the choice of associated u *LeadSurrogate* is ambiguous shall be associated with the nearest

possible u *LeadSurrogate* that would otherwise have no corresponding \u[*TrailSurrogate*.](#410)

[*LeadSurrogate*](#410) ::

[*Hex4Digits*](#170) but only if the SV of *Hex4Digits*  is in the inclusive range 0xD800 to 0xDBFF

[*TrailSurrogate*](#410) ::

[*Hex4Digits*](#170) but only if the SV of *Hex4Digits*  is in the inclusive range 0xDC00 to 0xDFFF

[*NonSurrogate*](#410) ::

[*Hex4Digits*](#170) but only if the SV of *Hex4Digits*  is not in the inclusive range 0xD800 to 0xDFFF

[*IdentityEscape*](#411)[U] ::

[+U] [*SyntaxCharacter*](#410)

[+U] /

[~U] [*SourceCharacter*](#158) but not [*UnicodeIDContinue*](#165)

[*DecimalEscape*](#411) ::

[*DecimalIntegerLiteral*](#167) [lookahead ∉ [*DecimalDigit*]](#167)

[*CharacterClassEscape*](#411) :: one of

d D s S w W

[*CharacterClass*](#411)[U] ::

[ [lookahead ∉ { ^ }]*ClassRanges*[?U] ]

[ ^*ClassRanges*[?U] ]

[*ClassRanges*](#411)[U] ::

[empty]

[*NonemptyClassRanges*](#411)[?U]

[*NonemptyClassRanges*](#411)[U] ::

[*ClassAtom*](#411)[?U]

[*ClassAtom*](#411)[?U]*NonemptyClassRangesNoDash*[?U]

[*ClassAtom*](#411)[?U] ‐*ClassAtom*[?U]*ClassRanges*[?U]

[*NonemptyClassRangesNoDash*](#411)[U] ::

[*ClassAtom*](#411)[?U]

[*ClassAtomNoDash*](#411)[?U]*NonemptyClassRangesNoDash*[?U]

[*ClassAtomNoDash*](#411)[?U] ‐*ClassAtom*[?U]*ClassRanges*[?U]

[*ClassAtom*](#411)[U] ::

‐

[*ClassAtomNoDash*](#411)[?U]

[*ClassAtomNoDash*](#411)[U] ::

[*SourceCharacter*](#158) but not one of \  or ]  or ‐

\*ClassEscape*[?U]

[*ClassEscape*](#411)[U] ::

[*DecimalEscape*](#411)

b

[+U] ‐

[*CharacterEscape*](#410)[?U]

[*CharacterClassEscape*](#411)

Annex B

Additional ECMAScript Features for Web Browsers

(normative)

The ECMAScript language syntax and semantics de伀氂ined in this annex are required when the ECMAScript host is a web

browser. The content of this annex is normative but optional if the ECMAScript host is not a web browser.

NOTE

This annex describes various legacy features and other characteristics of web browser based ECMAScript

implementations. All of the language features and behaviours speci伀氂ied in this annex have one or more

undesirable characteristics and in the absence of legacy usage would be removed from this speci伀氂ication.

However, the usage of these features by large numbers of existing web pages means that web browsers must

continue to support them. The speci伀氂ications in this annex de伀氂ine the requirements for interoperable

implementations of these legacy features.

These features are not considered part of the core ECMAScript language. Programmers should not use or

assume the existence of these features and behaviours when writing new ECMAScript code. ECMAScript

implementations are discouraged from implementing these features unless the implementation is part of a

web browser or is required to run the same legacy ECMAScript code that web browsers encounter.

B.1 Additional Syntax

B.1.1 Numeric Literals

The syntax and semantics of 11.8.3 is extended as follows except that this extension is not allowed for [strict mode code:](#159)

Syntax

[*NumericLiteral*](#557) ::

[*DecimalLiteral*](#167)

[*BinaryIntegerLiteral*](#167)

[*OctalIntegerLiteral*](#167)

[*HexIntegerLiteral*](#168)

[*LegacyOctalIntegerLiteral*](#557)

[*LegacyOctalIntegerLiteral*](#557) ::

0[*OctalDigit*](#167)

[*LegacyOctalIntegerLiteral*](#557)[*OctalDigit*](#167)

[*DecimalIntegerLiteral*](#557) ::

0

[*NonZeroDigit*](#167)*DecimalDigits*opt

[*NonOctalDecimalIntegerLiteral*](#557)

[*NonOctalDecimalIntegerLiteral*](#557) ::

0[*NonOctalDigit*](#557)

[*LegacyOctalLikeDecimalIntegerLiteral*](#557)[*NonOctalDigit*](#557)

[*NonOctalDecimalIntegerLiteral*](#557)[*DecimalDigit*](#167)

[*LegacyOctalLikeDecimalIntegerLiteral*](#557) ::

0[*OctalDigit*](#167)

[*LegacyOctalLikeDecimalIntegerLiteral*](#557)[*OctalDigit*](#167)

[*NonOctalDigit*](#557) :: one of

8 9

B.1.1.1 Static Semantics

The MV of *LegacyOctalIntegerLiteral* :: 0*OctalDigit*  is the MV of [*OctalDigit*.](#167)

The MV of *LegacyOctalIntegerLiteral* ::*LegacyOctalIntegerLiteral OctalDigit*  is (the MV of [*LegacyOctalIntegerLiteral*](#557)

times 8) plus the MV of [*OctalDigit*.](#167)

The MV of *DecimalIntegerLiteral* ::*NonOctalDecimalIntegerLiteral*  is the MV of [*NonOctalDecimalIntegerLiteral*.](#557)

The MV of *NonOctalDecimalIntegerLiteral* :: 0*NonOctalDigit*  is the MV of [*NonOctalDigit*.](#557)

The MV of *NonOctalDecimalIntegerLiteral* ::*LegacyOctalLikeDecimalIntegerLiteral NonOctalDigit*  is (the MV of

[*LegacyOctalLikeDecimalIntegerLiteral*](#557) times 10) plus the MV of [*NonOctalDigit*.](#557)

The MV of *NonOctalDecimalIntegerLiteral* ::*NonOctalDecimalIntegerLiteral DecimalDigit*  is (the MV of

[*NonOctalDecimalIntegerLiteral*](#557) times 10) plus the MV of [*DecimalDigit*.](#167)

The MV of *LegacyOctalLikeDecimalIntegerLiteral* :: 0*OctalDigit*  is the MV of [*OctalDigit*.](#167)

The MV of *LegacyOctalLikeDecimalIntegerLiteral* ::*LegacyOctalLikeDecimalIntegerLiteral OctalDigit*  is (the MV of

[*LegacyOctalLikeDecimalIntegerLiteral*](#557) times 10) plus the MV of [*OctalDigit*.](#167)

The MV of *NonOctalDigit* :: 8  is 8.

The MV of *NonOctalDigit* :: 9  is 9.

B.1.2 String Literals

The syntax and semantics of 11.8.4 is extended as follows except that this extension is not allowed for [strict mode code:](#159)

Syntax

[*EscapeSequence*](#558) ::

[*CharacterEscapeSequence*](#170)

[*LegacyOctalEscapeSequence*](#558)

[*HexEscapeSequence*](#170)

[*UnicodeEscapeSequence*](#170)

[*LegacyOctalEscapeSequence*](#558) ::

[*OctalDigit*](#167) [lookahead ∉ *OctalDigit*]

[*ZeroToThree*](#558)*OctalDigit* [lookahead ∉ [*OctalDigit*]](#167)

[*FourToSeven*](#558)[*OctalDigit*](#167)

[*ZeroToThree*](#558)*OctalDigit*[*OctalDigit*](#167)

[*ZeroToThree*](#558) :: one of

0 1 2 3

[*FourToSeven*](#558) :: one of

4 5 6 7

This de伀氂inition of *EscapeSequence* is not used in strict mode or when parsing [*TemplateCharacter*.](#174)

B.1.2.1 Static Semantics

The SV of *EscapeSequence* ::*LegacyOctalEscapeSequence*  is the SV of the [*LegacyOctalEscapeSequence*.](#558)

The SV of *LegacyOctalEscapeSequence* ::*OctalDigit*  is the code unit whose value is the MV of the [*OctalDigit*.](#167)

The SV of *LegacyOctalEscapeSequence* ::*ZeroToThree OctalDigit*  is the code unit whose value is (8 times the MV of the

[*ZeroToThree*](#558)) plus the MV of the [*OctalDigit*.](#167)

The SV of *LegacyOctalEscapeSequence* ::*FourToSeven OctalDigit*  is the code unit whose value is (8 times the MV of the

[*FourToSeven*](#558)) plus the MV of the [*OctalDigit*.](#167)

The SV of *LegacyOctalEscapeSequence* ::*ZeroToThree OctalDigit OctalDigit*  is the code unit whose value is (64 (that

is, 82) times the MV of the *ZeroToThree*) plus (8 times the MV of the 伀氂irst *OctalDigit*) plus the MV of the second [*OctalDigit*](#167)

.

The MV of *ZeroToThree* :: 0  is 0.

The MV of *ZeroToThree* :: 1  is 1.

The MV of *ZeroToThree* :: 2  is 2.

The MV of *ZeroToThree* :: 3  is 3.

The MV of *FourToSeven* :: 4  is 4.

The MV of *FourToSeven* :: 5  is 5.

The MV of *FourToSeven* :: 6  is 6.

The MV of *FourToSeven* :: 7  is 7.

B.1.3 HTML‑like Comments

The syntax and semantics of 11.4 is extended as follows except that this extension is not allowed when parsing source code

using the goal symbol [*Module*:](#308)

Syntax

[*Comment*](#559) ::

[*MultiLineComment*](#559)

[*SingleLineComment*](#163)

[*SingleLineHTMLOpenComment*](#559)

[*SingleLineHTMLCloseComment*](#559)

[*SingleLineDelimitedComment*](#559)

[*MultiLineComment*](#559) ::

/\**FirstCommentLine*opt*LineTerminator MultiLineCommentChars*opt \*/*HTMLCloseComment*opt

[*FirstCommentLine*](#559) ::

[*SingleLineDelimitedCommentChars*](#559)

[*SingleLineHTMLOpenComment*](#559) ::

<!‐‐*SingleLineCommentChars*opt

[*SingleLineHTMLCloseComment*](#559) ::

[*LineTerminatorSequence*](#162)[*HTMLCloseComment*](#559)

[*SingleLineDelimitedComment*](#559) ::

/\**SingleLineDelimitedCommentChars*opt \*/

[*HTMLCloseComment*](#559) ::

[*WhiteSpaceSequence*](#559)opt*SingleLineDelimitedCommentSequence*opt ‐‐>*SingleLineCommentChars*opt

[*SingleLineDelimitedCommentChars*](#559) ::

[*SingleLineNotAsteriskChar*](#559)*SingleLineDelimitedCommentChars*opt

\**SingleLinePostAsteriskCommentChars*opt

[*SingleLineNotAsteriskChar*](#559) ::

[*SourceCharacter*](#158) but not one of \*  or [*LineTerminator*](#162)

[*SingleLinePostAsteriskCommentChars*](#559) ::

[*SingleLineNotForwardSlashOrAsteriskChar*](#559)*SingleLineDelimitedCommentChars*opt

\**SingleLinePostAsteriskCommentChars*opt

[*SingleLineNotForwardSlashOrAsteriskChar*](#559) ::

[*SourceCharacter*](#158) but not one of /  or \*  or [*LineTerminator*](#162)

[*WhiteSpaceSequence*](#559) ::

[*WhiteSpace*](#162)*WhiteSpaceSequence*opt

[*SingleLineDelimitedCommentSequence*](#559) ::

[*SingleLineDelimitedComment*](#559)*WhiteSpaceSequence*opt*SingleLineDelimitedCommentSequence*opt

Similar to a *MultiLineComment* that contains a line terminator code point, a *SingleLineHTMLCloseComment* is considered to

be a *LineTerminator* for purposes of parsing by the syntactic grammar.

B.1.4 Regular Expressions Patterns

The syntax of 21.2.1 is modi伀氂ied and extended as follows. These changes introduce ambiguities that are broken by the

ordering of grammar productions and by contextual information. When parsing using the following grammar, each

alternative is considered only if previous production alternatives do not match.

This alternative pattern grammar and semantics only changes the syntax and semantics of BMP patterns. The following

grammar extensions include productions parameterized with the [U] parameter. However, none of these extensions change

the syntax of Unicode patterns recognized when parsing with the [U] parameter present on the goal symbol.

Syntax

[*Term*](#560)[U] ::

[+U] [*Assertion*](#560)[U]

[+U] [*Atom*](#562)[U]

[+U] [*Atom*](#562)[U][*QuantiᲪier*](#409)

[~U] [*QuantiᲪiableAssertion*](#560)[*QuantiᲪier*](#409)

[~U] [*Assertion*](#560)

[~U] [*ExtendedAtom*](#560)[*QuantiᲪier*](#409)

[~U] [*ExtendedAtom*](#560)

[*Assertion*](#560)[U] ::

^

$

\ b

\ B

[+U] ( ? =*Disjunction*[U] )

[+U] ( ? !*Disjunction*[U] )

[~U] [*QuantiᲪiableAssertion*](#560)

[*QuantiᲪiableAssertion*](#560) ::

( ? =*Disjunction* )

( ? !*Disjunction* )

[*ExtendedAtom*](#560) ::

.

\[*AtomEscape*](#561)

[*CharacterClass*](#411)

(*Disjunction* )

( ? :*Disjunction* )

[*InvalidBracedQuantiᲪier*](#560)

[*ExtendedPatternCharacter*](#560)

[*InvalidBracedQuantiᲪier*](#560) ::

{*DecimalDigits* }

{*DecimalDigits* , }

{*DecimalDigits* ,*DecimalDigits* }

[*ExtendedPatternCharacter*](#560) ::

[*SourceCharacter*](#158) but not one of ^ $ . \* + ? ( ) [ |

[*AtomEscape*](#561)[U] ::

[+U] [*DecimalEscape*](#411)

[+U] [*CharacterEscape*](#561)[U]

[+U] [*CharacterClassEscape*](#411)

[~U] [*DecimalEscape*](#411) but only if the integer value of *DecimalEscape*  is <= \_NcapturingParens\_

[~U] [*CharacterClassEscape*](#411)

[~U] [*CharacterEscape*](#561)

[*CharacterEscape*](#561)[U] ::

[*ControlEscape*](#410)

c[*ControlLetter*](#410)

[*HexEscapeSequence*](#170)

[*RegExpUnicodeEscapeSequence*](#410)[?U]

[~U] [*LegacyOctalEscapeSequence*](#558)

[*IdentityEscape*](#561)[?U]

[*IdentityEscape*](#561)[U] ::

[+U] [*SyntaxCharacter*](#410)

[+U] /

[~U] [*SourceCharacter*](#158) but not c

[*NonemptyClassRanges*](#561)[U] ::

[*ClassAtom*](#561)[?U]

[*ClassAtom*](#561)[?U]*NonemptyClassRangesNoDash*[?U]

[+U] [*ClassAtom*](#561)[U] ‐*ClassAtom*[U]*ClassRanges*[U]

[~U] [*ClassAtomInRange*](#561) ‐*ClassAtomInRange*[*ClassRanges*](#411)

[*NonemptyClassRangesNoDash*](#561)[U] ::

[*ClassAtom*](#561)[?U]

[*ClassAtomNoDash*](#561)[?U]*NonemptyClassRangesNoDash*[?U]

[+U] [*ClassAtomNoDash*](#561)[U] ‐*ClassAtom*[U]*ClassRanges*[U]

[~U] [*ClassAtomNoDashInRange*](#561) ‐*ClassAtomInRange*[*ClassRanges*](#411)

[*ClassAtom*](#561)[U] ::

‐

[*ClassAtomNoDash*](#561)[?U]

[*ClassAtomNoDash*](#561)[U] ::

\*ClassEscape*[?U]

[*SourceCharacter*](#158) but not one of ]  or ‐

[*ClassAtomInRange*](#561) ::

‐

[*ClassAtomNoDashInRange*](#561)

[*ClassAtomNoDashInRange*](#561) ::

\[*ClassEscape*](#561)

[*SourceCharacter*](#158) but not one of ]  or ‐

[*ClassEscape*](#561)[U] ::

b

[+U] [*DecimalEscape*](#411)

[+U] [*CharacterEscape*](#561)[U]

[+U] [*CharacterClassEscape*](#411)

[+U] ‐

[~U] [*DecimalEscape*](#411) but only if the integer value of *DecimalEscape*  is 0

[~U] [*CharacterClassEscape*](#411)

[~U] c[*ClassControlLetter*](#562)

[~U] [*CharacterEscape*](#561)

[*ClassControlLetter*](#562) ::

[*DecimalDigit*](#167)

\_

NOTE

When the same left hand sides occurs with both [+U] and [~U] guards it is to control the disambiguation

priority.

B.1.4.1 Pattern Semantics