

---

Ortopediafuncionaldelosmaxilareswilmasimoespdfdownload ((LINK))

[Download](#)

---

A: I'm not sure why the "È~ êĚ" regex would not match, however in that case, you can just match with `^`. `std::regex reg("^.*");` `std::string input = ...` if `(reg.search(input) != std::regex_constants::no_match)` { ... } A: Unfortunately, `std::regex` is pretty darn bad at parsing HTML. You'll end up having to parse the DOM yourself, for example: You can use C++ DOM and create a document fragment from a DOM HTML string. You'll then need to parse the DOM HTML and find the element you care about. Here is a way you can do that: `#include "boost/spirit/home/unicode.hpp"` `#include #include #include //using the spirit::char_ class //with boost::spirit::x3::char_ makes DOM expression much faster using boost::spirit::x3::char_;` `struct MyParse : boost::static_visitor { MyParse(const boost::locale::id_t& l) : loc(l) {} template void visit(T const& node) { if(node.nodeType() == EOF) throw std::runtime_error("Document End: " + node.nodeName()); std::string name = node.nodeName(); if(name.length() > 0 && name[0] == '>') name = name.substr(1); typename boost::container::std_vector result; for(const auto& child :` `node.children)`

